

日 本 国 特 許 庁
JAPAN PATENT OFFICE

J1040 U.S. PRO
09/940982
08/29/01

別紙添付の書類に記載されている事項は下記の出願書類に記載されて
る事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed
with this Office

出 願 年 月 日

Date of Application:

2001年 2月22日

出 願 番 号

Application Number:

特願2001-046250

出 願 人

Applicant(s):

株式会社日立製作所

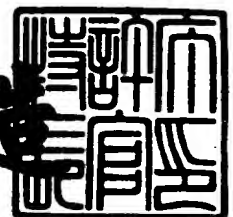
U.S. Appln. Filed 8-29-01
Inventor: T. Endo et al
Mattingly Stanger & Molur
Docket NIT-295

CERTIFIED COPY OF
PRIORITY DOCUMENT

2001年 7月27日

特 許 庁 長 官
Commissioner,
Japan Patent Office

及 川 耕 造



出証番号 出証特2001-3066026

【書類名】 特許願

【整理番号】 NT00P0641

【提出日】 平成13年 2月22日

【あて先】 特許庁長官 殿

【国際特許分類】 G06K 19/073

【発明者】

【住所又は居所】 東京都国分寺市東恋ヶ窪一丁目280番地 株式会社日立製作所 中央研究所内

【氏名】 遠藤 隆

【発明者】

【住所又は居所】 東京都国分寺市東恋ヶ窪一丁目280番地 株式会社日立製作所 中央研究所内

【氏名】 神永 正博

【発明者】

【住所又は居所】 東京都国分寺市東恋ヶ窪一丁目280番地 株式会社日立製作所 中央研究所内

【氏名】 渡邊 高志

【発明者】

【住所又は居所】 東京都国分寺市東恋ヶ窪一丁目280番地 株式会社日立製作所 中央研究所内

【氏名】 大木 優

【特許出願人】

【識別番号】 000005108

【氏名又は名称】 株式会社日立製作所

【代理人】

【識別番号】 100068504

【弁理士】

【氏名又は名称】 小川 勝男

【電話番号】 03-3661-0071

【選任した代理人】

【識別番号】 100086656

【弁理士】

【氏名又は名称】 田中 恭助

【電話番号】 03-3661-0071

【選任した代理人】

【識別番号】 100094352

【弁理士】

【氏名又は名称】 佐々木 孝

【電話番号】 03-3661-0071

【手数料の表示】

【予納台帳番号】 081423

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 情報処理装置

【特許請求の範囲】

【請求項 1】 プログラムを格納するプログラム格納部、データを保存するデータ格納部を有する記憶装置と

プログラムにしたがい、所定の処理を実行する中央演算装置とを有し、

プログラムは、中央演算装置の指示を与える処理命令から構成される一つ以上のデータ処理手段からなり、

一つのデータ処理手段が、その入力データを処理し、処理済データを出力する入力データ処理手段を含み、且つ

生成される攪乱用データ X_i のハミングウエイトが常に一定値であり、且つ、生成された X_i を前記入力データ処理手段で処理済攪乱用データ X_o に変換した後のハミングウエイトも常に一定になるような、攪乱用データ X_i を生成する攪乱用データ生成手段を有し、

前記攪乱用データ生成手段は、

前記攪乱用データ X_i を使って入力データ D_1 を変形し、変形データ H_1 を作成するデータ変形手段と、

変形データ H_1 を前記入力データ処理手段と同じ処理を行い、処理済変形データ H_2 を得る変形データ処理手段と、

前記攪乱用データ X_i に前記入力データ処理手段と同じ処理を行い、処理済攪乱用データ X_o を生成する攪乱用データ処理手段と、

処理済攪乱用データ X_o を使用して処理済変形データ H_2 を処理し、入力データ D_1 を入力データ処理手段で処理した結果である処理済データ D_2 を得るデータ逆変形処理手段と、を有することを特徴とする情報処理装置。

【請求項 2】 生成される攪乱用データ X_i のハミングウエイトが、常に一定値であり、且つ、生成された X_i を予め定義された入力データ処理に従って処理して生成した処理済攪乱用データ X_o のハミングウエイトも常に一定であるような、攪乱用データおよび処理済攪乱用データ生成手段であって、

攪乱用データおよび処理済攪乱用データを生成する手段は、ハミングウエイト

一定の乱数を生成する攪乱用データ生成手段と、攪乱用データ処理手段と、及びハミングウエイト検査手段とを有し、

前記攪乱用データ生成手段で生成されたデータを攪乱用データ処理手段で処理し、処理結果をハミングウエイト検査手段でハミングウエイトが所定の値であるかを検査し、所定の値でない場合に、攪乱用データ生成手段のデータ生成からやり直すように制御信号を送るハミングウエイト検査手段を有し、ハミングウエイトが一定で、攪乱用データ処理手段で処理した後も、ハミングウエイトが一定となるような攪乱用データを生成するように構成された、攪乱用データおよび処理済攪乱用データ生成手段を、有する情報処理装置。

【請求項 3】 前記攪乱用データおよび処理済攪乱用データを生成する手段が、あらかじめ、ハミングウエイトが一定でかつ、攪乱用データ処理手段で処理を行った結果も、ハミングウエイトが一定となるようなデータを選択して複数格納した攪乱用データ格納手段と、攪乱用データ格納手段に格納されたデータをランダムに選択する攪乱用データ選択手段と、選択された攪乱用データ X_i を処理して、処理済の攪乱用データを生成する攪乱用データ処理をなす手段を有することを特徴とした、請求項 1 に記載の情報処理装置。

【請求項 4】 前記攪乱用データおよび処理済攪乱用データを生成する手段として、あらかじめ、ハミングウエイトが一定でかつ、攪乱用データ処理手段で処理を行った結果もハミングウエイトが一定となるようなデータを、攪乱用データ処理手段で処理した結果と組み合わせて、複数組格納した攪乱用データおよび処理済攪乱用データ格納手段と、攪乱用データおよび処理済攪乱用データ格納手段に格納されたデータをランダムに選択する攪乱用データおよび処理済攪乱用データ選択手段とを有することを特徴とした、請求項 1 に記載の情報処理装置。

【請求項 5】 前記攪乱用データおよび処理済攪乱用データ格納手段に格納するデータの組の数が偶数個であり、且つ攪乱用データおよび処理済攪乱用データのいずれのビットについても、0 となるか 1 となるかの確率が 0.5 となるように選択したデータを格納している攪乱用データおよび処理済攪乱用データ格納手段を有することを特徴とする、請求項 4 に記載の情報処理装置。

【請求項 6】 ハミングウエイト一定の乱数を生成する手段として、発生する乱

数のビット数の半分のビット数の乱数を発生させる乱数発生手段と、ビット反転演算を計算するビット反転演算手段と、乱数発生手段で生成されたデータと、ビット反転演算手段の計算結果を結合して、所定のビット数の数値を生成する、データ結合手段から成る、ハミングウエイト一定乱数生成手段を、有することを特徴とする情報処理装置。

【請求項 7】 ハミングウエイト一定の乱数を生成する手段として、乱数発生手段と、前記乱数発生手段により生成された乱数のハミングウエイト計算手段と、前記ハミングウエイト計算手段の結果を検査し、目標ハミングウエイトと等しくない場合に乱数発生手段に対して乱数の再生成を行わせる、ハミングウエイト検査手段とを有することを特徴とした、ハミングウエイト一定乱数生成手段を、有することを特徴とする情報処理装置。

【請求項 8】 ハミングウエイト一定の乱数を生成する手段として、発生させたい乱数のビット数の約数となるビット数ハミングウエイト一定の乱数を発生させるハミングウエイト一定部分乱数発生手段と、発生させたい乱数のビット数に達するまで、前記ハミングウエイト一定部分乱数発生手段に乱数を生成させる乱数発生制御手段と、発生した乱数を結合して、所定のビット数の乱数とするデータ結合手段からなる、ハミングウエイト一定乱数生成手段を、有することを特徴とする情報処理装置。

【請求項 9】 プログラムを格納するプログラム格納部、データを保存するデータ格納部を有する記憶装置と

プログラムにしたがい、所定の処理を実行する中央演算装置を有し、プログラムは、中央演算装置の指示を与える処理命令から構成される一つ以上のデータ処理手段を有し、

一つのデータ処理手段が、インデックスとそれに対応したデータから成る表を有し、入力データをインデックスとして表引きを行い、インデックスに対応したデータを処理済データとして出力する入力データ処理手段を有し、

ハミングウエイトの値が常に一定となる第 1 の攪乱用データ $X1i$ と、

ハミングウエイトの値が常に一定となる第 2 の攪乱用データ $X2i$ と、

第 1 攪乱用データ $X1i$ でインデックスを攪乱し、

第 2 攪乱用データ $X2i$ でインデックスに対応したデータを攪乱することにより生成された変形済み表を使い、

前記攪乱用データ Xi を使って入力データ $D1$ を変形し、変形データ $H1$ を作成するデータ変形手段と、

変形データ $H1$ をインデックスとして、前記変形済み表を引いて変形データ $H2$ を取り出す変形済み表アクセス処理手段と、

変形データ $H2$ を第 2 攪乱用データ $X2i$ を用いて、

入力データ $D1$ を入力データ処理手段で処理した結果である処理済データ $D2$ を得るデータ逆変形処理手段とを有する、ことを特徴とする情報処理装置。

【請求項 10】 第 1 攪乱用データ $X1i$ および第 2 攪乱用データ $X2i$ および変形済み表を生成する手段として、第 1 攪乱用データ $X1i$ を生成するための第 1 ハミングウエイト一定乱数生成手段と、第 2 攪乱用データ $X2i$ を生成するための第 2 ハミングウエイト一定乱数生成手段と、第 1 攪乱用データ $X1i$ と第 2 攪乱用データ $X2i$ と表を用いて、表のインデックスを第 1 攪乱用データ $X1i$ で変形し、表のデータを第 2 攪乱用データ $X2i$ で変形し、変形済み表を生成する表変形手段を有すること、を特徴とする請求項 9 に記載の情報処理装置。

【請求項 11】 第 1 攪乱用データ $X1i$ および第 2 攪乱用データ $X2i$ および変形済み表を生成する手段として、ハミングウエイト一定の値を予め複数格納している第 1 攪乱用データ格納手段と、

第 1 攪乱用データ格納手段に格納されたデータをランダムに選択し、第 1 攪乱用データ $X1i$ とするための第 1 攪乱用データ選択手段と、

ハミングウエイト一定の値を予め複数格納している第 2 攪乱用データ格納手段と、第 2 攪乱用データ格納手段に格納されたデータをランダムに選択し、第 2 攪乱用データ $X2i$ とするための第 2 攪乱用データ選択手段と、

第 1 攪乱用データ $X1i$ と第 2 攪乱用データ $X2i$ と表を用いて、表のインデックスを第 1 攪乱用データ $X1i$ で変形し、表のデータを第 2 攪乱用データ $X2i$ で変形し、変形済み表を生成する表変形手段を有することを特徴とする、請求項 9 に記載の情報処理装置。

【請求項 12】 第 1 攪乱用データ $X1i$ および第 2 攪乱用データ $X2i$ および変形済

み表を生成する手段として、第 1 攪乱用データになりうる、ハミングウェイト一定の値と、第 2 攪乱用データになりうる、ハミングウェイト一定の値と前記第 1 攪乱用データとなり得る値と前記第 2 攪乱用データとなり得る値と表を用いて、あらかじめ表のインデックスを第 1 攪乱用データ $X1i$ で変形し、表のデータを第 2 攪乱用データ $X2i$ で変形して作成した変形済み表を、作成したときに用いた第 1 攪乱用データと第 2 攪乱用データを組にしたものを、複数組格納した第一攪乱用データ、第 2 攪乱用データおよび変形済み表格納手段と、

第一攪乱用データ、第 2 攪乱用データおよび変形済み表格納手段からランダムに、

第 1 攪乱用データと第 2 攪乱用データと変形済み表を選択する、第 1 攪乱用データ第 2 攪乱用データおよび変形済み表選択手段を有することを特徴とする、請求項 9 に記載の情報処理装置。

【請求項 1 3】 プログラムを格納するプログラム格納部、データを保存するデータ格納部を有する記憶装置と

プログラムにしたがい、所定の処理を実行する中央演算装置を有し、プログラムは、中央演算装置の指示を与える処理命令から構成される一つ以上のデータ処理手段からなり、

一つのデータ処理手段が、表引きを行い、表引き結果をデータ処理処理し、処理結果を処理済データとして出力する入力データ処理手段を有し、且つ、

ハミングウェイトの値が常に一定となる第 1 の攪乱用データ $X1i$ と、ハミングウェイトの値が常に一定となり、かつ前記表引き後に行われるデータ処理を行った結果のハミングウェイトも常に一定となる様な第 2 の攪乱用データ $X2i$ と、前記第 2 の攪乱用データ $X2i$ をデータ処理した結果の処理済第 2 攪乱用データ $X2i$ と、

第 1 攪乱用データ $X1i$ でインデックスを攪乱し、

第 2 攪乱用データ $X2i$ でインデックスに対応したデータを攪乱することにより生成された変形済み表を使い、

前記攪乱用データ Xi を使って入力データ Di を変形し、変形データ $H1$ を作成するデータ変形手段と、

変形データH1をインデックスとして、前記変形済み表を引いて変形データH2を取り出す変形済み表アクセス処理手段と、

処理済変形データH2を処理し、処理済変形データH3を得る変形済みデータ処理手段と、

処理済み変形データH3を処理済第2攪乱用データX2oを用いて、入力データD1を用いて表引きを行い、表引き結果をデータ処理処理した結果である処理済データD2を得るデータ逆変形処理手段とを有することを特徴とする情報処理装置。

【請求項14】 第1攪乱用データX1iおよび第2攪乱用データX2iおよび処理済第2攪乱用データX2oおよび変形済み表を生成する手段として、

第1攪乱用データX1iを生成するための第1ハミングウエイト一定乱数生成手段と、

第2攪乱用データX2iを生成するための第2ハミングウエイト一定乱数生成手段と、

第2攪乱用データX2iを処理し、処理済第2攪乱用データX2oを生成する攪乱用データ処理手段と、

処理済第2攪乱用データX2oのハミングウエイトを計算し、処理済第2攪乱用データX2oのハミングウエイトが不適切な場合は第2ハミングウエイト一定乱数生成手段に対して、

第2攪乱用データX2iの再生成を行わせる、ハミングウエイト検査手段と、

第1攪乱用データX1iと第2攪乱用データX2iと表を用いて、表のインデックスを第1攪乱用データX1iで変形し、表のデータを第2攪乱用データX2iで変形し、変形済み表を生成する表変形手段とを有することを特徴とする、請求項13に記載の情報処理装置。

【請求項15】 前記第1攪乱用データX1iおよび第2攪乱用データX2iおよび処理済第2攪乱用データX2oおよび変形済み表を生成する手段として、

ハミングウエイト一定の値を予め複数格納している第1攪乱用データ格納手段と、

第1攪乱用データ格納手段に格納されたデータをランダムに選択し、第1攪乱

用データX1iとするための第1 攪乱用データ選択手段と、

ハミングウエイト一定かつ攪乱用データ処理手段で処理した結果のハミングウエイトも一定となる様な値を複数格納している第2 攪乱用データ格納手段と、第2 攪乱用データ格納手段に格納されたデータをランダムに選択し、第2 攪乱用データX2iとするための第2 攪乱用データ選択手段と、

第2 攪乱用データX2iを処理し、処理済第2 攪乱用データX2oを生成する攪乱用データ処理手段と、

第1 攪乱用データX1iと第2 攪乱用データX2iと表を用いて、表のインデックスを第1 攪乱用データX1iで変形し、表のデータを第2 攪乱用データX2iで変形し、変形済み表を生成する表変形手段とを有することを特徴とする、請求項1 3に記載の情報処理装置。

【請求項1 6】 前記第1 攪乱用データX1iおよび処理済第2 攪乱用データX2oおよび変形済み表を生成する手段として、

ハミングウエイト一定の値を予め複数格納している第1 攪乱用データ格納手段と

第1 攪乱用データ格納手段に格納されたデータをランダムに選択し、第1 攪乱用データX1iとするための第1 攪乱用データ選択手段と、

ハミングウエイト一定かつ攪乱用データ処理手段で処理した結果のハミングウエイトも一定となる様な、第2 攪乱用データと処理済第2 攪乱用データの組を複数組格納している第2 攪乱用データおよび処理済第2 攪乱用データ格納手段と、第2 攪乱用データおよび処理済第2 攪乱用データ格納手段に格納された、第2 攪乱用データと処理済第2 攪乱用データの組をランダムに選択し、第2 攪乱用データX2iおよび処理済第2 攪乱用データX2oとするための第2 攪乱用データおよび処理済第2 攪乱用データ選択手段と、

第1 攪乱用データX1iと第2 攪乱用データX2iと表を用いて、表のインデックスを第1 攪乱用データX1iで変形し、表のデータを第2 攪乱用データX2iで変形し、変形済み表を生成する表変形手段と、を有することを特徴とする、請求項1 3に記載の情報処理装置。

【請求項1 7】 前記第1 攪乱用データX1iおよび処理済第2 攪乱用データX2oお

よび変形済み表を生成する手段として、

ハミングウエイト一定の第 1 攪乱用データに使用できる値と、

ハミングウエイト一定かつ攪乱用データ処理手段で処理した結果のハミングウエイトも一定となる様な、第 2 攪乱用データに使用できる値を処理した結果選られる処理済第 2 攪乱用データに使用できる値と、

第 1 攪乱用データに使用できる値と前記処理済第 2 攪乱用データに使用できる値を生成したときに用いた第 2 攪乱用データに使用できる値を用いて、表のインデックスを第 1 攪乱用データ $X1i$ に使用できる値で変形し、表のデータを第 2 攪乱用データ $X2i$ に使用できる値で変形し、変形済み表を生成し、

前記第 1 攪乱用データに使用できる値と処理済第 2 攪乱用データに使用できる値と変形済み表の組を複数格納した、第 1 攪乱用データおよび処理済第 2 攪乱用データおよび変形済み表格納手段と、

第 1 攪乱用データと処理済第 2 攪乱用データと変形済み表の組をランダムに選択し、

第 1 攪乱用データ $X1i$ と処理済第 2 攪乱用データ $X2o$ と処理済表とするための、第 1 攪乱用データおよび処理済第 2 攪乱用データおよび変形済み表選択手段とを有することを特徴とする、請求項 13 に記載の情報処理装置。

【請求項 18】 プログラムを格納するプログラム格納部、データを保存するデータ格納部を有する記憶装置と

プログラムにしたがい、所定の処理を実行し、データ処理を行なう中央演算装置を有し、

プログラムは、中央演算装置の指示を与える処理命令から構成される一つ以上のデータ処理手段を有し、

一つのデータ処理手段が、表引きを行い、表引き結果をデータ処理処理し、処理結果を処理済データとして出力する入力データ処理手段を含み、前記処理手段を複数回繰り返すことで処理結果を生成し、且つ、

ハミングウエイトの値が常に一定となる第 1 の攪乱用データ $X1i$ と、ハミングウエイトの値が常に一定となり、かつ前記表引き後に行われるデータ処理を行った結果のハミングウエイトも常に一定となる様な第 2 の攪乱用データ $X2$

iと、前記第2の攪乱用データX2iをデータ処理した結果の処理済第2攪乱用データX2oと、第1攪乱用データX1iでインデックスを攪乱し、第2攪乱用データX2iでインデックスに対応したデータを攪乱することにより生成された変形済み表を使い、

前記攪乱用データXiを使って入力データD1を変形し、変形データH1を作成するデータ変形手段と、

変形データH1をインデックスとして、前記変形済み表を引いて変形データH2を取り出す変形済み表アクセス処理手段と、

処理済変形データH2を処理し、処理済変形データH3を得る変形済みデータ処理手段と、

第1攪乱用データX1iを用いて処理済み変形データH3を変形し、処理済変形データH4を得るデータ変形処理手段と、

処理済第2攪乱用データX2oを用いて処理済み変形データH4を変形し、処理済変形データH5を得るデータ変形処理手段と、

処理済変形データH5をインデックスとして、前記変形済み表を引いて変形データH6を取り出す変形済み表アクセス処理手段と、

処理済変形データH6を処理し、処理済変形データH7を得る変形済みデータ処理手段と、

処理済第2攪乱用データX2oを用いて処理済み変形データH7を変形し、入力データD1を用いて表引きを行い、表引き結果をデータ処理処理し、さらにその結果を用いて表引きを行い、表引き結果をデータ処理処理した結果である処理済データD2を得るデータ逆変形処理手段と、を有することを特徴とする情報処理装置。

【請求項19】 第1攪乱用データX1iおよび処理済第2攪乱用データX2oおよび変形済み表を生成する手段として、

第1攪乱用データX1iを生成するための第1ハミングウェイト一定乱数生成手段と、

第2攪乱用データX2iを生成するための第2ハミングウェイト一定乱数生成手段と、

第 2 攪乱用データ X2i を処理し、処理済第 2 攪乱用データ X2o を生成する攪乱用データ処理手段と、

処理済第 2 攪乱用データ X2o のハミングウェイトを計算し、処理済第 2 攪乱用データ X2o のハミングウェイトが不適切な場合は第 2 ハミングウェイト一定乱数生成手段に対して、

第 2 攪乱用データ X2i の再生成を行わせる、ハミングウェイト検査手段と、

第 1 攪乱用データ X1i と第 2 攪乱用データ X2i と表を用いて、表のインデックスを第 1 攪乱用データ X1i で変形し、表のデータを第 2 攪乱用データ X2i で変形し、変形済み表を生成する表変形手段とを有することを特徴とする、請求項 18 に記載の情報処理装置。

【請求項 20】 前記第 1 攪乱用データ X1i および処理済第 2 攪乱用データ X2o および変形済み表を生成する手段として、

ハミングウェイト一定の値を予め複数格納している第 1 攪乱用データ格納手段と

第 1 攪乱用データ格納手段に格納されたデータをランダムに選択し、第 1 攪乱用データ X1i とするための第 1 攪乱用データ選択手段と、

ハミングウェイト一定かつ攪乱用データ処理手段で処理した結果のハミングウェイトも一定となる様な値を複数格納している第 2 攪乱用データ格納手段と、第 2 攪乱用データ格納手段に格納されたデータをランダムに選択し、第 2 攪乱用データ X2i とするための第 2 攪乱用データ選択手段と、

第 2 攪乱用データ X2i を処理し、処理済第 2 攪乱用データ X2o を生成する攪乱用データ処理手段と、

第 1 攪乱用データ X1i と第 2 攪乱用データ X2i と表を用いて、表のインデックスを第 1 攪乱用データ X1i で変形し、表のデータを第 2 攪乱用データ X2i で変形し、変形済み表を生成する表変形手段とを有することを特徴とする、請求項 18 に記載の情報処理装置。

【請求項 21】 第 1 攪乱用データ X1i および処理済第 2 攪乱用データ X2o および変形済み表を生成する手段として、

ハミングウェイト一定の値を予め複数格納している第 1 攪乱用データ格納手段と

第 1 攪乱用データ格納手段に格納されたデータをランダムに選択し、第 1 攪乱用データ $X1i$ とするための第 1 攪乱用データ選択手段と、

ハミングウエイト一定かつ攪乱用データ処理手段で処理した結果のハミングウエイトも一定となる様な、第 2 攪乱用データと処理済第 2 攪乱用データの組を複数組格納している第 2 攪乱用データおよび処理済第 2 攪乱用データ格納手段と、第 2 攪乱用データおよび処理済第 2 攪乱用データ格納手段に格納された、第 2 攪乱用データと処理済第 2 攪乱用データの組をランダムに選択し、第 2 攪乱用データ $X2i$ および処理済第 2 攪乱用データ $X2o$ とするための第 2 攪乱用データおよび処理済第 2 攪乱用データ選択手段と、

第 1 攪乱用データ $X1i$ と第 2 攪乱用データ $X2i$ と表を用いて、表のインデックスを第 1 攪乱用データ $X1i$ で変形し、表のデータを第 2 攪乱用データ $X2i$ で変形し、変形済み表を生成する表変形手段とを有することを特徴とする、請求項 18 に記載の情報処理装置。

【請求項 22】 第 1 攪乱用データ $X1i$ および処理済第 2 攪乱用データ $X2o$ および変形済み表を生成する手段として、

ハミングウエイト一定の第 1 攪乱用データに使用できる値と、

ハミングウエイト一定かつ攪乱用データ処理手段で処理した結果のハミングウエイトも一定となる様な、第 2 攪乱用データに使用できる値を処理した結果選られる処理済第 2 攪乱用データに使用できる値と、

第 1 攪乱用データに使用できる値と前記処理済第 2 攪乱用データに使用できる値を生成したときに用いた第 2 攪乱用データに使用できる値を用いて、表のインデックスを第 1 攪乱用データ $X1i$ に使用できる値で変形し、表のデータを第 2 攪乱用データ $X2i$ に使用できる値で変形し、変形済み表を生成し、

前記第 1 攪乱用データに使用できる値と処理済第 2 攪乱用データに使用できる値と変形済み表の組を複数格納した、第 1 攪乱用データおよび処理済第 2 攪乱用データおよび変形済み表格納手段と、

第 1 攪乱用データと処理済第 2 攪乱用データと変形済み表の組をランダムに選択し、第 1 攪乱用データ $X1i$ と処理済第 2 攪乱用データ $X2o$ と処理済表とするため

の、第 1 攪乱用データおよび処理済第 2 攪乱用データおよび変形済み表選択手段とを有することを特徴とする、請求項 1 8 に記載の情報処理装置。

【請求項 2 3】 プログラムを格納するプログラム格納部、データを保存するデータ格納部を有する記憶装置と

プログラムにしたがい、所定の処理を実行し、データ処理を行なう中央演算装置を有し、

プログラムは、中央演算装置の指示を与える処理命令から構成される一つ以上のデータ処理手段からなり、

一つのデータ処理手段が、表引きを行い、表引き結果をデータ処理し、処理結果を処理済データとして出力する入力データ処理手段を含み、前記処理手段を複数回繰り返すことで処理結果を生成し、且つ、

ハミングウエイトの値が常に一定である第 1 の攪乱用データ $X1i$ と、

ハミングウエイトの値が常に一定であり、かつ前記表引き後に行われるデータ処理を行った結果のハミングウエイトも常に一定となる様な第 2 の攪乱用データ $X2i$ と、前記第 2 の攪乱用データ $X2i$ をデータ処理した結果の処理済第 2 攪乱用データ $X2o$ と、

ハミングウエイトの値が常に一定である第 3 の攪乱用データ $X3i$ と、

ハミングウエイトの値が常に一定であり、かつ前記表引き後に行われるデータ処理を行った結果のハミングウエイトも常に一定となる様な第 4 の攪乱用データ $X4i$ と、前記第 4 の攪乱用データ $X4i$ をデータ処理した結果の処理済第 4 攪乱用データ $X4o$ と、

第 1 攪乱用データ $X1i$ を用いてインデックスを攪乱し、

さらに第 3 攪乱用データ $X3i$ で前記第一攪乱用データを用いた攪乱されたインデックスをさらに攪乱し、

第 2 攪乱用データ $X2i$ を用いて前記インデックスに対応したデータを攪乱し、

第 4 攪乱用データ $X4i$ を用いて、前記第 2 攪乱用データを用いて攪乱されたデータをさらに攪乱することにより生成された第 2 変形済み表を使い、

前記第 3 攪乱用データ $X3i$ を使って入力データ $D1$ を変形し、変形データ $H1$ を作成するデータ変形手段と、

前記第 1 攪乱用データ X1i を使って変型データ H1 を変形し、変形データ H2 を作成するデータ変形手段と、

変形データ H2 をインデックスとして、前記第 2 変形済み表を引いて変形データ H3 を取り出す変形済み表アクセス処理手段と、

処理済変形データ H3 を処理し、処理済変形データ H4 を得る変形済みデータ処理手段と、

第 3 攪乱用データ X3i を用いて処理済み変形データ H4 を変形し、処理済変形データ H5 を得るデータ変形処理手段と、

第 1 攪乱用データ X1i を用いて処理済み変形データ H5 を変形し、処理済変形データ H6 を得るデータ変形処理手段と、

処理済第 2 攪乱用データ X2o を用いて処理済み変形データ H6 を変形し、処理済変形データ H7 を得るデータ変形処理手段と、

処理済第 4 攪乱用データ X4o を用いて処理済み変形データ H7 を変形し、処理済変形データ H8 を得るデータ変形処理手段と、

処理済変形データ H8 をインデックスとして、前記第 2 変形済み表を引いて変形データ H9 を取り出す変形済み表アクセス処理手段と、

処理済変形データ H9 を処理し、処理済変形データ H10 を得る変形済みデータ処理手段と、

処理済第 2 攪乱用データ X2o を用いて処理済み変形データ H10 を逆変形し、変型データ H11 を得るデータ逆変型処理手段と、

処理済第 4 攪乱用データ X4o を用いて処理済み変形データ H11 を逆変形し、

入力データ D1 を用いて表引きを行い、表引き結果をデータ処理処理し、さらにその結果を用いて表引きを行い、表引き結果をデータ処理処理した結果である処理済データ D2 を得るデータ逆変形処理手段とを有することを特徴とする情報処理装置。

【請求項 24】 第 1 攪乱用データ X1i および処理済第 2 攪乱用データ X2o および第 3 攪乱用データ X3i および処理済第 4 攪乱用データ X4o および第 2 変形済み表を生成する手段として、

第 1 攪乱用データ X1i を生成するための第 1 ハミングウェイト一定乱数生成手

段と、

第 2 攪乱用データ X_{2i} を生成するための第 2 ハミングウェイト一定乱数生成手段と、

第 2 攪乱用データ X_{2i} を処理し、処理済第 2 攪乱用データ X_{2o} を生成する攪乱用データ処理手段と、

処理済第 2 攪乱用データ X_{2o} のハミングウェイトを計算し、処理済第 2 攪乱用データ X_{2o} のハミングウェイトが不適切な場合は第 2 ハミングウェイト一定乱数生成手段に対して、

第 2 攪乱用データ X_{2i} の再生成を行わせる、ハミングウェイト検査手段と、

第 1 攪乱用データ X_{1i} と第 2 攪乱用データ X_{2i} と表を用いて、表のインデックスを第 1 攪乱用データ X_{1i} で変形し、表のデータを第 2 攪乱用データ X_{2i} で変形し、変形済み表を生成する表変形手段と、

第 3 攪乱用データ X_{3i} を生成するための第 1 乱数生成手段と、

第 4 攪乱用データ X_{4i} を生成するための第 2 乱数生成手段と、

第 4 攪乱用データ X_{4i} を処理し、処理済第 4 攪乱用データ X_{4o} を生成する攪乱用データ処理手段と、

第 3 攪乱用データ X_{3i} と第 4 攪乱用データ X_{4i} と変形済み表とを用いて、変形済み表のインデックスを第 3 攪乱用データ X_{3i} で変形し、変形済み表のデータを第 4 攪乱用データ X_{4i} で変形し、第 2 変形済み表を生成する表変形処理手段とを有することを特徴とする、請求項 2 3 に記載の情報処理装置。

【請求項 2 5】 第 1 攪乱用データ X_{1i} および処理済第 2 攪乱用データ X_{2o} および第 3 攪乱用データ X_{3i} および処理済第 4 攪乱用データ X_{4o} および第 2 変形済み表を生成する手段として、

ハミングウェイト一定の値を予め複数格納している第 1 攪乱用データ格納手段と、

第 1 攪乱用データ格納手段に格納されたデータをランダムに選択し、第 1 攪乱用データ X_{1i} とするための第 1 攪乱用データ選択手段と、

ハミングウェイト一定かつ攪乱用データ処理手段で処理した結果のハミングウェイトも一定となる様な値を複数格納している第 2 攪乱用データ格納手段と、第

2 攪乱用データ格納手段に格納されたデータをランダムに選択し、第 2 攪乱用データ X_{2i} とするための第 2 攪乱用データ選択手段と、

第 2 攪乱用データ X_{2i} を処理し、処理済第 2 攪乱用データ X_{2o} を生成する攪乱用データ処理手段と、

第 1 攪乱用データ X_{1i} と第 2 攪乱用データ X_{2i} と表を用いて、表のインデックスを第 1 攪乱用データ X_{1i} で変形し、表のデータを第 2 攪乱用データ X_{2i} で変形し、変形済み表を生成する表変形手段と、

第 3 攪乱用データ X_{3i} を生成するための第 1 乱数生成手段と、

第 4 攪乱用データ X_{4i} を生成するための第 2 乱数生成手段と、

第 4 攪乱用データ X_{4i} を処理し、処理済第 4 攪乱用データ X_{4o} を生成する攪乱用データ処理手段と、

第 3 攪乱用データ X_{3i} と第 4 攪乱用データ X_{4i} と変形済み表とを用いて、変形済み表のインデックスを第 3 攪乱用データ X_{3i} で変形し、変形済み表のデータを第 4 攪乱用データ X_{4i} で変形し、第 2 変形済み表を生成する表変形処理手段とを有することを特徴とする、請求項 2 3 に記載の情報処理装置。

【請求項 2 6】 第 1 攪乱用データ X_{1i} および処理済第 2 攪乱用データ X_{2o} および第 3 攪乱用データ X_{3i} および処理済第 4 攪乱用データ X_{4o} および第 2 変形済み表を生成する手段として、

ハミングウエイト一定の値を予め複数格納している第 1 攪乱用データ格納手段と、

第 1 攪乱用データ格納手段に格納されたデータをランダムに選択し、第 1 攪乱用データ X_{1i} とするための第 1 攪乱用データ選択手段と、

ハミングウエイト一定かつ攪乱用データ処理手段で処理した結果のハミングウエイトも一定となる様な、第 2 攪乱用データと処理済第 2 攪乱用データの組を複数組格納している第 2 攪乱用データおよび処理済第 2 攪乱用データ格納手段と、第 2 攪乱用データおよび処理済第 2 攪乱用データ格納手段に格納された、第 2 攪乱用データと処理済第 2 攪乱用データの組をランダムに選択し、第 2 攪乱用データ X_{2i} および処理済第 2 攪乱用データ X_{2o} とするための第 2 攪乱用データおよび処理済第 2 攪乱用データ選択手段と、

第 1 攪乱用データ $X1i$ と第 2 攪乱用データ $X2i$ と表を用いて、表のインデックスを第 1 攪乱用データ $X1i$ で変形し、表のデータを第 2 攪乱用データ $X2i$ で変形し、変形済み表を生成する表変形手段と、

第 3 攪乱用データ $X3i$ を生成するための第 1 乱数生成手段と、

第 4 攪乱用データ $X4i$ を生成するための第 2 乱数生成手段と、

第 4 攪乱用データ $X4i$ を処理し、処理済第 4 攪乱用データ $X4o$ を生成する攪乱用データ処理手段と、

第 3 攪乱用データ $X3i$ と第 4 攪乱用データ $X4i$ と変形済み表とを用いて、変形済み表のインデックスを第 3 攪乱用データ $X3i$ で変形し、変形済み表のデータを第 4 攪乱用データ $X4i$ で変形し、第 2 変形済み表を生成する表変形処理手段とを有することを特徴とする、請求項 2 3 に記載の情報処理装置。

【請求項 2 7】 プログラムを格納するプログラム格納部、データを保存するデータ格納部を有する記憶装置と

プログラムにしたがい、所定の処理を実行し、データ処理を行なう中央演算装置を有し、

プログラムは、中央演算装置の指示を与える処理命令から構成される一つ以上のデータ処理手段を有し、

一つのデータ処理手段が、メッセージと秘密鍵を入力とし、Data Encryption Standard (DES) 暗号処理を行い、暗号化処理を行ない、前記暗号化処理の処理結果を生成し、且つ、

平文を攪乱する平文攪乱用データ PX を用いて、メッセージを変形する手段と、

秘密鍵を攪乱する秘密鍵攪乱用データ KX を用いて、秘密鍵データ K を変形する手段と、

DES 暗号処理で用いられる $SBOX$ 表のインデックスを攪乱するための $SBOX$ アドレス攪乱用データ $SinX1$ を用いて $SBOX$ 表のインデックスを変形し $SBOX$ 表を並べ替え、かつ $SBOX$ 表の出力を攪乱するための $SBOX$ データ攪乱用データ $SoutX$ とを用いて並べ替えられた $SBOX$ 表の出力を攪乱し、変形済み $SBOX$ 表を作成するための $SBOX$ 表変形手段と、

$SBOX$ の表引き処理の直前の XOR 結果が、データが $SinX1$ および平文

攪乱用データ PX もしくは PX を変形した値で変形された値となるように調整するための逆変形処理もしくは変形処理を、 XOR の 2 つの入力データのいずれかもしくは両方に有し、

$SBOX$ の表引き処理の前までにデータが $SinX1$ で変形された値に調整するための、逆変形処理を $SBOX$ の直前に有し、

DES の最終ラウンド終了後の IP 置換処理の直前もしくは直後に、平文攪乱用データ PX もしくは PX を変形された値を逆変形するための逆変形処理手段を有することを特徴とする、情報処理装置。

【請求項 28】 前記 $SBOX$ 表アドレス攪乱用データ $SinX1$ のハミングウェイトが一定かつ、 $SBOX$ 表データ攪乱用データ $SoutX$ のハミングウェイトと、転置済み $SBOX$ データ攪乱用データ $XSoutX$ のハミングウェイトが一定であるような、 $SoutX$ を用いていることを特徴とする、請求項 27 に記載の情報処理装置。

【請求項 29】 前記 $SBOX$ 表アドレス攪乱に 2 つ以上の攪乱用データを用いて、アドレスの変形を複数回おこない、

$SBOX$ 表のデータ攪乱に 2 つ以上の攪乱用データを用いて、変形を複数回以上行なう $SBOX$ 表変形手段を有することを特徴とする、請求項 27 に記載の情報処理装置。

【請求項 30】 $SBOX$ 表アドレス攪乱用データのうち、1 つ以上の攪乱用データのハミングウェイトのが一定かつ、 $SBOX$ 表データ攪乱用データのうち、1 つ以上の攪乱用データのハミングウェイトが一定であるような、攪乱用データを用いることを特徴とする、請求項 29 に記載の情報処理装置。

【請求項 31】 ハミングウェイトが一定となる攪乱用データが、中央演算装置で一度に処理できるビット長に分割した際のそれぞれの部分ビットにおけるハミングウェイトも一定であるような、 $SBOX$ アドレス攪乱用データ $SinX1$ 、 $SBOX$ データ攪乱用データ $SoutX1$ を用いることを特徴とする、請求項 30 に記載の情報処理装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、機密性の高いＩＣカードなどの耐タンパ装置に関するものである。

【０００２】

【従来の技術】

ＩＣカードは、勝手に書き換えることが許されないような個人情報の保持や、秘密情報である暗号鍵を用いたデータの暗号化や、或いは暗号文の復号化を行う装置である。ＩＣカード自体は電源を持っておらず、ＩＣカード用のリーダライタに差し込まれると、電源の供給を受け、動作可能となる。ＩＣカードは動作可能になると、リーダライタから送信されるコマンドを受信し、そのコマンドに従って、データの転送等の処理を行う。

【０００３】

ＩＣカードの基本概念は、図１に示すように、カード１０１の上に、ＩＣカード用チップ１０２を搭載したものである。図に示すように、一般にＩＣカードは、所定位置に、供給電圧端子Ｖｃｃ、グランド端子ＧＮＤ、リセット端子ＲＳＴ、入出力端子Ｉ／Ｏ、クロック端子ＣＬＫ及び、を有する。この端子の位置は、ＩＳ０７８１６の規格に定められている。これらの諸端子を通して、リーダライタから電源の供給やリーダライタとのデータの通信を行う。こうしたＩＣカードを用いた通信に関しては、例えばＷ．Ｒankl And Effing：ＳＭＡＲＴＣＡＲＤ　ＨＡＮＤＢＯＯＫ、Ｊohn Wiley & Sons s、１９９７、ＰＰ．４１などに見られる。

【０００４】

ＩＣカードに搭載される半導体チップの構成は、基本的には通常のマイクロコンピュータと同じ構成である。図２はＩＣカードに搭載される半導体チップの基本的構成を示すブロック図である。図２に見られるように、カード部材用の半導体チップは、中央処理装置（ＣＰＵ）２０１、記憶装置２０４、入出力（Ｉ／Ｏ）ポート２０７、コ・プロセッサ２０２を有する。システムによってはコ・プロセッサはない場合もある。ＣＰＵ２０１は、論理演算や算術演算などを行う装置であり、記憶装置２０４は、プログラムやデータを格納する装置である。入出力ポートは、リーダライタと通信を行う装置である。コ・プロセッサは、暗号処理

そのもの、または、暗号処理に必要な演算を高速に行う装置である。これには、例えば、RSA暗号の剰余演算を行う為の特別な演算装置や、DES暗号のラウンド処理を行う暗号装置などがある。ICカード用プロセッサの中には、コ・プロセッサを持たないものも多くある。データバス203は、各装置を接続するバスである。

【0005】

記憶装置204は、ROM (Read Only Memory) やRAM (Random Access Memory)、EEPROM (Electric Erasable Programmable Read Only Memory) などを有する。ROMは、記憶情報を変更できないメモリであり、主にプログラムを格納するメモリである。RAMは自由に書き換えができるメモリであるが、電源の供給が中断されると、記憶している内容は消滅する。ICカードがリーダーライタから抜かれると電源の供給が中断されるため、RAMの内容は、保持されなくなる。EEPROMは、電源の供給が中断されてもその内容を保持することができるメモリである。このEEPROMは記憶情報を書き換える必要があり、ICカードがリーダーライタから抜かれても、保持が可能なデータを格納するために使われる。例えば、プリペイドカードでのプリペイドの度数などは、使用するたびに書き換えられ、かつリーダーライタが抜かれてもデータを保持する必要があるため、EEPROMに保持される。

・ 【0006】

ICカードは、プログラムや重要な情報がICカード用チップの中に密閉されているため、重要な情報を格納したり、カードの中で暗号処理を行うために用いられる。従来、ICカードでの暗号を解読する難しさは、暗号アルゴリズムの解読の困難さと同じと考えられていた。しかし、ICカードが暗号処理を行っている時の消費電流を観測し、解析することにより、暗号アルゴリズムの解読より容易に暗号処理の内容や暗号鍵が推定される可能性が示唆されている。消費電流は、リーダーライタから供給されている電流を測定することにより観測することができ、この攻撃法の詳細は、例えば、John Wiley & Sons社、W. Rankle & W. Fffing著「SMART CARD HA

NDBOOK」の8.5.1.1 Passive Protective Mechanisms (263ページ)にこのような危険性が記載されている。それは次のような理由による。ICカード用チップを構成しているCMOSは、出力状態が1から0あるいは0から1に変わった時に電流を消費する。特に、データバス203においては、バスドライバーの電流や、配線及び、配線に接続されているトランジスタの静電容量のため、バスの値が1から0あるいは0から1に変わると、大きな電流が流れる。そのため、消費電流を観測すれば、ICカード用チップの中で、何が動作しているか分かる可能性を示唆している。

【0007】

図3は、ICカード用チップの1サイクルでの消費電流の波形を示したものである。処理しているデータに依存して、電流波形が301や302のように異なる。このような差は、バス203を流れるデータや中央演算装置201で処理しているデータに依存して生じる。

【0008】

16ビットのプリチャージバスにデータを転送する場合を考える。プリチャージバスは、データ転送の前に全てのビットを"0"にそろえるバスである。このバスに、値は違うが、"1"のビットの数が同じデータ、たとえば、"1"のビットの数が2である16進数の"88"と"11"を転送した場合、電流波形はほぼ同じ波形になる。この理由は、"0"から"1"へ変化したビットの数が同じであるため、同じように電流を消費し、同じ電流波形になったからである。もし、"1"のビットの数が1つ異なるデータ、たとえば"1"のビットの数が3である"89"や"19"を転送した場合、"1"のビットの数が2のデータとは消費電流波形が異なる。これは、3ビット分のバスの値が"0"から"1"に変化したため、その分の電流が消費される。そのため、先の2ビットが変化したデータに比べて、消費電流が1ビット分大きくなる。一般に、"1"のビットの数が多いほど、電流波形は高くなるという規則性がある。この規則性から、転送されるデータを推定することができる。

【0009】

具体的な命令でどのように差が出るかを、次のような左シフト命令を例にして

説明する。

logical_shiftl R1 (式1)

この命令は、レジスタ1の内容を左シフトし、最上位ビットの値をコンディションコードレジスタのキャリーフラグに入れる命令である。レジスタR1の最上位ビットが内部バスを經由してコンディションコードレジスタに転送されるため、電流波形の大きさを比較すれば、最上位ビットの"0"と"1"が識別できる可能性がある。もしR1に重要なデータが入っていれば、そのデータの1ビットが"0"であるか"1"であるかが分かる可能性がある。特にDESのような暗号処理では、暗号鍵をシフトする操作が頻発する。このシフト操作の時に暗号鍵のデータを推定できる電流波形が生じ、暗号鍵を推定される危険性がある。

【0010】

これは、コ・プロセッサ202の演算でも同じである。演算内容が暗号鍵に依存した偏りがあると、その偏りが消費電流から求められ、暗号鍵が推定される可能性がある。

【0011】

特願平10-354156号の「情報処理装置、端タンパ処理装置」では、この課題を解決する方法として、攪乱用データで処理データを変形し、変形されたデータを処理し、処理結果を攪乱用データを用いて逆変換することにより、処理中の消費電流とデータの関連性を減らすことで解決しようとしている。

【0012】

次の命令列を例として、問題点を説明する。

logical_rotatel R1 (式2)

xor R1 R2 (式3)

(式2)では、レジスタR1の値を左に論理ローテートし、レジスタR1に格納する。(式3)では、レジスタR2とR1の排他的論理和を取り、R2に格納する。(式2)と(式3)では、処理するデータをそのまま扱っているため、データの内容によって消費電流波形の大きさが変わり、電流波形を観測することによって、データが推定できる。

特開平10-354156号では、問題を解決するために、勝手に選んだ乱数X1、X2

を攪乱用データとして用い、以下に示すように（式4）、（式5）でR1、R2に格納されたデータを変形し、（式6）、（式7）で変形データを処理し、R2に格納する。（式8）、（式9）で逆変形するための準備計算を行い、（式10）でR2に格納された処理済変形データを逆変形し、（式2）（式3）で得られるものと同じ値がR2に格納される。

$$\text{xor} \quad X1 \quad R1 \quad (\text{式4})$$

$$\text{xor} \quad X2 \quad R2 \quad (\text{式5})$$

$$\text{logical_rotetel} \quad R1 \quad (\text{式6})$$

$$\text{xor} \quad R1 \quad R2 \quad (\text{式7})$$

$$\text{logical_rotetel} \quad X1 \quad (\text{式8})$$

$$\text{xor} \quad X1 \quad X2 \quad (\text{式9})$$

$$\text{xor} \quad X2 \quad R2 \quad (\text{式10})$$

ここで問題となるのは、特開平10-354156号では、データのハミングウェイトが直接観測されないように、攪乱用データを使っていた。しかし、攪乱用データはある確率でハミングウェイトが0や8（特別な値）になる。もし、攪乱用データがそのような特別のハミングウェイトを持つと、実際の処理データのハミングウェイトが直接観測できることになる。本発明では、攪乱用データのハミングウェイトを0や8などの特別な値に成ることを防ぐものである。

【0013】

具体的には、（式4）、（式5）を計算する際に、X1、X2の値に依存した消費電流の差が観測可能であり、X1、X2のハミングウェイトが推定できる。たとえば、X1、X2のハミングウェイトと消費電流が比例するプロセッサの場合、ハミングウェイトが0となる場合を検出できる。同様に、xor演算でのビットがする反転数と消費電流が比例する場合においても、反転するビット数は、X1、X2のハミングウェイトと等しい。ハミングウェイトが0となる値は0以

外にはありえないため、消費電流を観測することで、攪乱用データが0となる場合の測定データのみを識別することが出来る。上記の攪乱方式では、(式6)(式7)の計算に際して、(式2)(式3)と同じ電流波形が観測されることとなり、アタックが可能となる。

【0014】

【発明が解決しようとしている課題】

本願発明は、高いセキュリティを持つカード部材などの耐タンパー情報処理装置を提供するものである。

【0015】

本願発明の技術的な課題は、カード部材、例えばICカード用チップでのデータ処理と消費電流との関連性を減らすことである。消費電流とチップの処理との関連性が減れば、観測した消費電流の波形からICカード用チップ内での処理や暗号鍵を推測することが困難になる。即ち、本願発明は、カード部材等に高いセキュリティを持たせんとするものである。

【0016】

【課題を解決するための手段】

本願発明の着眼点は、ICカード用チップで消費される電流値と、処理されているデータの関連性を減らすための方法として、処理するデータを攪乱用データで変形し、データの処理を変形したデータで処理し、処理後に攪乱用データを用いて逆変換し、正しい処理結果を求めるものである。さらに、攪乱用データと消費電流の関連性を減らすために、攪乱用データとして、データを変形する際に用いる攪乱用データのバイナリ表現した際のハミングウエイトが常に一定値となりかつ各ビットが"0"と"1"のどちらの値をとるのかの確率が0.5となる。且つ、逆変換する際に用いる攪乱用データのバイナリ表現もまたハミングウエイトが一定となり、各ビットが"0"と"1"のどちらの値をとるのかの確率が0.5であるような攪乱用データを生成することにより、攪乱用データを用いた処理の消費電流と攪乱データの関連性を減らし、攪乱用データを消費電流から推定した後、変形したデータを消費電流から推定し、推定された攪乱用データと変形したデータから、元の処理データを推定するアタックを困難にする。尚、ここで、

ハミングウエイトとは、二進数における“1”の数を意味する。

【0017】

さらに、攪乱用データを攪乱用データの生成方法として、あらかじめ攪乱データとして用いることの出来る値を複数生成し、格納しておくことにより、攪乱データを生成する際に攪乱用データを生成する際の消費電流と攪乱用データの関連性を減らし、攪乱用データを推定することを困難にする。

【0018】

【発明の実施の形態】

以下、本発明の実施例に付いて図面を参照しながら説明する。

【0019】

図1はICカードの概観を示したものである。ICカード101は、ISO7816の規格により、大きさや、ICカードのチップ102の位置や接点の数および割り当てなどが規定されている。

【0020】

図2はICカードチップ102の内部構成である。構成に付いては、従来技術の説明で既に述べた通りである。本発明は、プログラム205で処理するデータに攪乱を加えることにより、処理中に生じるICカード用チップのハードウェアが消費する電流波形から、本物のデータの推定を困難にさせるものである。

【0021】

従来の技術で説明したように、データをそのまま処理した場合、消費電流を測定することにより、データを推定することが可能である。また、処理されるデータを攪乱用のデータを用いて変形し、変形したデータに対して処理を行い、選ばれた処理結果を攪乱用に用いたデータもしくは攪乱用に用いたデータを処理した値を用いて逆変形を行い、本物の処理後データと等しい値を得ることにより、処理中に消費される電流値とデータとの間の相関を減らし、消費電流の測定からデータを推定することを困難にする従来技術では、攪乱に用いるデータについて制限を設けていないため、攪乱用データが処理させる際の消費電流値により、攪乱用データを推定し、測定データを分類することで、なおアタックが可能である。

【0022】

例えば、XORを攪乱用の関数として用いた場合、攪乱用データの全ビットが1の場合や全ビットが0の場合などは、消費電流により識別可能であり、また識別率が100%でない場合でも、多くの測定サンプルの平均値を計算することにより、識別誤りの影響をデータの推定には影響が無くなるようにすることが出来る。

【 0 0 2 3 】

尚、前記処理の例は、例えば、ローテート、シフト、ビット位置置換、拡大ビット位置置換などの処理をあげる事が出来る。

【 0 0 2 4 】

そこで、攪乱用データのハミングウェイトが攪乱用データのビット長の半分に等しくなり、かつ攪乱用データのそれぞれのビット位置での0と1の出現確率が0.5となるように攪乱用データを生成することで、攪乱用データが処理される際の消費電流より、攪乱用データが容易に特定できなくなる。

【 0 0 2 5 】

入力データをD1、処理を関数f、出力データをD2とするとき、

$$D2 = f(D1) \quad (\text{式} 11)$$

となる。処理fの電流波形を測定することで、D1が推定可能とする。ここで、攪乱用データX1iを用いて、D1を攪乱する変形関数hと、hの逆変換を行うgという変換関数があり、次の(式12)もしくは(式13)の関係を満たす場合、(式11)の代わりに(式12)もしくは(式13)を計算することで、(式11)相当の処理を行う。

【 0 0 2 6 】

(式12)もしくは(式13)のいずれが使用できるのかは、関数f、関数hの性質に依存する。(式12)を満たす、関数f、関数hの例としては、関数fがローテート、シフト、ビット位置置換などの処理(式14)であり、関数hがxor(式15)の場合が例としてあげられる。その場合は、関数gもまた、xor(式16)となる。

【 0 0 2 7 】

一方、(式 1 3) を満たす、関数 f 、関数 h の例としては、関数 f が加減算で、関数 h も加減算の場合、関数 f が乗除算で、関数 h も乗除算の場合が例として上げられる。

(式 1 2) や (式 1 3) においても、処理 f の電流波形を測定することで、 $h(D1, X1i)$ の値は推定可能であるが、 $X1i$ が推定できないと、 $D1$ を復元することはできない。

$$f(D1) = g(f(h(D1, X1i)), f(X1i)) \quad (\text{式 1 2})$$

$$f(D1) = g(f(h(D1, X1i)), X1i) \quad (\text{式 1 3})$$

$$h(x, y) = x \text{ xor } y \quad (\text{式 1 4})$$

$$f(x) = \text{rotate_right}(x) \quad (\text{式 1 5})$$

$$g(x, y) = x \text{ xor } y \quad (\text{式 1 6})$$

ところが、 $X1i$ が値が特定の値 C になった場合に、外部からの観測で識別可能で、変形関数 h が既知である場合、 $h(D1, C)$ の値から、 h の逆関数を取ることによって、 $D1$ を復元出来る。外部からの電流測定により識別が容易な $X1i$ としては、すべてのビットが 0 である場合や、すべてのビットが 1 の場合などがある。というのは、ハミングウェイトが 0 となるデータは 0 以外には存在せず、同様にすべてのビットが 1 となる場合のハミングウェイトと等しいハミングウェイトを与える値は、すべてのビットが 1 となるもの以外にはないからである。関数 h として、 xor を用いた場合、 $X1i$ が 0 となった場合、 $h(D1, 0)$ が即 $D1$ の値と等しくなる。値のバラエティーが最も多いハミングウェイトは、データのビット長の半分のハミングウェイトである。

【 0 0 2 8 】

図 4 は、1 個の攪乱用データを用いた実施例である。本実施例の特徴は、攪乱用データのハミングウェイトを与えることによって、攪乱用データの各ビットが

オール 0 になったり、オール 1 になったりすることにより、データが推定されることを防ぐものである。データ変形処理手段 (4 0 2) で入力データ D 1 (4 0 1) を攪乱用データ X 1 i (4 0 3) を用いて変形し、変形データ H 2 (4 0 4) を生成し、変形データ処理手段 (4 0 5) にて、変形データ H 1 (4 0 4) を処理し、変形データ H 2 (4 0 6) を得る。データ逆変換処理手段 (4 0 7) で、変形データ H 2 (4 0 7) を変形済み攪乱用データ X 1 o (4 0 8) を用いて逆変換し、処理済データ D 2 (4 0 9) を得る。ここで、攪乱用データ X 1 i (4 0 3) と変形済み攪乱用データ X 1 o (4 0 8) は、ともにハミングウェイトが一定の値である。

【 0 0 2 9 】

この実施例は、請求項 1 に対応する実施例である。攪乱用データ X 1 i (4 0 3) と、変形済み攪乱用データ X 1 o (4 0 8) を作成する方法には幾つかの方法が有る。

【 0 0 3 0 】

図 5 は、ハミングウェイト一定の攪乱用データ X 1 i (5 0 2) およびハミングウェイト一定の処理済攪乱データ X 1 o (5 0 4) を生成する一実施例である。ハミングウェイト一定乱数生成手順 (5 0 1) により、ハミングウェイト一定の乱数を生成し、第 1 攪乱用データ X 1 i (5 0 2) とし、攪乱用データ処理手段 (5 0 3) で第 1 攪乱用データ X 1 i (5 0 2) を処理し、処理済攪乱用データ X 1 o (5 0 4) を得る。ハミングウェイト検査手段 (5 0 5) により、処理済攪乱用データ X 1 o (5 0 4) のハミングウェイトが検査され、所定のハミングウェイトと異なる場合は、ハミングウェイト一定の乱数生成手段 (5 0 1) に対して再生成制御信号を送り、攪乱用データ X 1 i (5 0 2) の生成からやりなおす。これは、請求項 2 の実施例である。前記ハミングウェイト検査は、具体的には多くは CPU で実行される。ハミングウェイト一定の乱数生成手段 (5 0 1) は CPU あるいはその生成手段 (Generator) がその役割を果たす。

【 0 0 3 1 】

ハミングウェイト一定乱数生成の方法はいくつか存在する。図 6 は、ハミング

ウェイト一定の乱数の生成方法の 1 実施例で、請求項 6 の実施例でもある。ここでは、発生したい乱数のビット数を $2n$ ビットとする。まず n ビット乱数発生手段 (601) により n ビット乱数 (602) を生成する。この n ビット乱数発生手段 (601) は、疑似乱数でもよいし、物理現象を測定して選られる真の乱数であってもよい。つぎにビット反転処理手段 (603) を用いて、発生した n ビット乱数 (602) を反転し、反転 n ビット乱数 (604) を生成する。つぎに、データ統合手段 (605) にて、 n ビット乱数 (602) と反転 n ビット乱数 (604) を結合して、ハミングウェイト一定 $2n$ ビット乱数 (606) を生成する。 n ビット乱数 (602) の各ビットのうち、値が 1 であるビット数を n_1 個、0 であるビット数を n_2 個とすると、

$$n_1 + n_2 = n \quad (\text{式 17})$$

であり、反転 n ビット乱数 (604) は n ビット乱数 (602) の反転値であるので、値が 1 であるビットの数は n_2 個、0 であるビット数が n_1 個となる。従って、 n ビット乱数 (602) と反転 n ビット乱数 (604) を結合して作成した乱数のハミングウェイトは、 $n_1 + n_2$ となり、(式 17) に示されるように、ここで生成される乱数のハミングウェイトは、常に n ビットとなる。

【0032】

図 7 はハミングウェイト一定乱数生成の別の実施例で、請求項 7 の実施例である。目標のハミングウェイト H を受け取り (702)、乱数 R を発生させる (703)。乱数 R のハミングウェイト RH を計算し (704)、発生させた乱数 R のハミングウェイト RH が目標のハミングウェイト H と等しいか比較し、等しくない場合は、乱数 R 発生 (703) からやり直す。乱数 R のハミングウェイトと、目標のハミングウェイト H が等しい場合は、乱数 R を結果として返し (706)、処理を終了する (707)。

【0033】

図 10 は、ハミングウェイト一定乱数生成の別の実施例で、請求項 8 の実施例である。予め、 m ビット長のデータで、ハミングウェイトが一定であるデータを

収めたテーブルを用意しておく。発生できるハミングウエイト一定の乱数は、 m の倍数ビット長にかぎられる。まず、発生したい乱数のビット長を n に設定し（1002）、 L に n を m で割った値を代入する（1004）。処理の基本的な流れは、 m ビット長のハミングウエイト一定の乱数を L 個生成し結合することで、 n ビット長のハミングウエイト一定の乱数を生成する。つぎに、ハミングウエイト一定の乱数の生成結果を収める D を0に初期化する（1004）。乱数 R を生成し、乱数 R をインデックスとして、あらかじめハミングウエイトが一定であるデータを収めたテーブルから値を1つ取り出し、 d に代入する（1006）。 D を m ビット分左シフトし、 d を加える（1007）。乱数 R 発生（1005）から D を m ビット分左シフトし、 d を加える処理（1007）までを、 L 回繰り返す（1008）、 D を結果として返す（1009）。

【0034】

図8は、決められたビット長（MaxBit）のデータで、ハミングウエイトがHammingビット（一定のハミングウエイト）となるデータをすべてリストアップし、 dat という配列に入れるための処理手順の実施例である。 dat 配列の大きさは、 $(MaxBitの階乗) / \{ (Hammingの階乗)^2 \}$ になる。尚、ここで、 dat 配列とは、ハミングウエイトが一定のデータを格納するための配列である。たとえば、8ビット長のデータで、ハミングウエイトが4ビットとなるようなデータは、70個になる。基本的な考え方は、Hamming個の1となるビット位置を格納する配列を用意し、その配列におさめられたビット位置が重ならないように1となるビット位置を変えて行くことで、すべての組み合わせを見つけ出すというものである。まず、ハミングウエイトをHammingに代入する（802）。つぎに生成したいデータのビット長をMaxBitに代入する（803）。次にHamming個の要素を持つ、1となるビットの位置を格納する pos 配列を0からHamming-1までの値で初期化する（804）。結果を格納するための dat 配列へのインデックス num を0に初期化する（805）。尚、ここで、 num は配列 dat のインデックス値で、処理（806）で生成される値を格納するインデックスの位置を示す。まず、 pos 配列に収められたビット位置から、データを計算し、 dat 配列のインデックスが num の位置に収める（806）。インデックス num に1を加える（807）。更新

するビット位置を決めるインデックス b を 0 に初期化する (808)。 b が一番上位に位置するビットにたどり着いているかチェックし (809)、たどり着いていなければ、処理 801 に、たどり着いていれば、処理 812 に分岐する。処理 801 では、1 ビット上位がすでに 1 になっているかチェックし、すでに 1 があれば、処理 812 に飛び、1 がなければ、処理 811 へ移行する。処理 811 では、ビットの位置を 1 ビット分上位方向に進め、処理 806 に移る。処理 812 では、現在注目しているビットが、1 となっているビットのうちの最上位のビットかどうかをチェックし、最上位であった場合には、処理 813 に進み、それ以外の場合は処理 814 に進む。処理 813 では、1 ビット上位にビット移動できるかチェックし、移動できる場合は、811 に飛び、移動できない場合は、処理 814 に飛ぶ。処理 814 では、現在注目しているビットが値が 1 となっているうちの最下位のビットかどうかをチェックし、最下位のビットであれば処理 815 に飛び、最下位ではない場合は処理 816 に飛ぶ。処理 816 では、下位側のビットのうち、1 となっているビットで一番近くにあるビットから数えて、1 ビット分上位ビットの位置に移動させる。処理 815 では、最下位にビットを移動させ、処理 817 に移る。処理 817 では、注目するビットを 1 つ分、上位方向に位置するビットに移す。処理 818 では、全部の組み合わせについて処理が終了したかをチェックし、まだ終わっていなければ、処理 806 に飛び、すべての組み合わせに付いて処理が終わっていれば、終了する (819)。結果は、 dat 配列に格納され、 dat 配列に格納されたデータの個数は、 num に設定される。

【0035】

図 11 は、攪乱用データ $X1i$ (1103)、変形済み攪乱用データ $X1o$ (1105) を生成する一実施例で、請求項 3 の実施例である。予め攪乱用データ $X1i$ として使用可能なデータを格納した攪乱用データ格納手段 (1102) より、攪乱用データ選択手段 (1101) がデータを選択し、攪乱用データ $X1i$ (1103) とする。選択された攪乱用データ $X1i$ (1103) を攪乱用データ処理手段 (1104) により処理し、変形済み攪乱用データ $X1o$ (1105) を生成する。攪乱用データ格納手段は通例、例えば RAM あるいはレジスタなどが、又攪乱用データ処理手段は通例、例えば CPU あるいは ALU などが用い

られる。図 9 は、攪乱用データ格納手段（1 1 0 2）に格納するための、攪乱用データを予め選択する方法の一実施例である。

【0 0 3 6】

図 9 の実施例は、図 8 の実施例と同様の手順で、ハミングウエイト一定のデータをリストアップしている。図 8 の実施例との違いは、リストアップされたデータをそのまま使用するのではなく、さらに攪乱用データ処理手段で処理し、処理結果のハミングウエイトを計算して $hxdat$ に代入し（9 0 7）、 $hxdat$ の値もハミングウエイト一定となることをチェックし（9 0 8）、一定の場合だけ、 dat 配列に格納するようにしている点である。その他の処理に付いては、図 8 と同一である。

【0 0 3 7】

図 1 2 は、攪乱用データ $X1i$ （1 2 0 3）、変形済み攪乱用データ $X1o$ （1 2 0 4）を生成する一実施例で、請求項 4 の実施例である。予め、図 9 の実施例や図 5 の実施例などの方法で、攪乱用データおよび処理済攪乱用データ格納手段（1 2 0 1）に攪乱用データおよび処理済攪乱用データの組を複数組格納し、攪乱用データおよび処理済攪乱用データ選択手段（1 2 0 2）により、攪乱用データおよび処理済攪乱用データ格納手段（1 2 0 1）から、攪乱用データ $X1i$ （1 2 0 3）及び処理済攪乱用データ $X1o$ （1 2 0 4）を取り出す。図 5 5 に図 1 2 に対応するテーブルの例を示す。これはデータ処理が、左ローテートの場合で、前記攪乱用データおよび処理済攪乱用データ格納手段（1 2 0 1）に格納されたデータ $X1i$ 、 $X1o$ を例示する。

【0 0 3 8】

また、攪乱用データおよび処理済攪乱用データ格納手段（1 2 0 1）に格納するデータの個数を偶数にし、格納するデータを適切に選択することで、請求項 5 の実施例となる。請求項 5 の実施例としては、最低限 2 組のデータがあればよい。

【0 0 3 9】

図 1 3 は、データ処理がテーブルルックアップで定義された処理を、2 個の攪乱データを用いて攪乱する実施例で、請求項 9 の一実施例である。入力データ

D1 (1301) を用いて、表を引き、処理済データ D2 (1310) 得る。表を Table とすると、D1 と D2 の関係は、(式19) のようになる。

$$D2 = \text{Table}[D1] \quad (\text{式19})$$

この表引き処理を行う際の電流を測定すると、D1 や D2 の値が推定できる。そこで、第1攪乱用データ X1i と第2攪乱用データ X2i と、表のインデックスと出力結果を変形する関数 f と関数 g を考え、変形済みの表 XTable を (式20) のように定義する。

$$\begin{aligned} \text{XTable}[f(i, X1i)] \\ = g(\text{Table}[i], X2i) \end{aligned} \quad (\text{式20})$$

また、g の逆関数を h と定義する。

$$D = h(g(D, X), X) \quad (\text{式22})$$

すると、表引き処理は、

$$H1 = f(D1, X1i) \quad (\text{式23})$$

$$H2 = \text{XTable}[H1] \quad (\text{式24})$$

$$D2 = h(H2, X2i) \quad (\text{式25})$$

となる。関数 f (x, y) は、x の値が異なる場合に、必ず異なる値になる必要がある。一方、関数 g、関数 h は、(式26) の関係を満たす必要がある。

$$a = h(g(a, X), X) \quad (\text{式26})$$

ここで、(式24) の処理の消費電流により H1 もしくは H2 が推定されても、

それぞれ $X1_i$ 、 $X2_i$ により攪乱されているため、(式 24) の測定結果だけでは、 $D1$ 、 $D2$ の値を推定することは出来ない。図 13 では、(式 23) にあたる処理が、第 1 攪乱用データ $X1_i$ (1303) を用いて、入力データ $D1$ (1301) をデータ変形処理手段 (1302) で処理し、変形データ $H1$ (1304) を得る処理に相当する。また、(式 24) に当たる処理は、変形データ $H1$ (1304) をインデックスとして、変形済み表 (1306) を表引きする変形済み表アクセス手段 (1305) を用いて、変形データ $H2$ (1307) を得る処理に相当する。(式 25) に相当する処理は、変形データ $H2$ (1307) を第 2 攪乱用データ $X2_i$ (1309) を用いてデータ逆変形処理手段 (1308) を用いて逆変形し、処理済データ $D2$ (1301) を得る処理に相当する。

【0040】

図 14 は、図 13 の実施例で用いる、第 1 攪乱用データ $X1_i$ (1403) および第 2 攪乱用データ $X2_i$ (1404) および変形済み表 (1407) を作成するための一実施例で、請求項 10 の一実施例である。第 1 ハミングウェイト一定乱数生成手段 (1401) により、第 1 攪乱用データ $X1_i$ (1403) を生成し、第 2 ハミングウェイト一定乱数生成手段 (1402) により、第 2 攪乱用データ $X2_i$ (1404) を生成し、表格納手段 (1405) に格納された、(式 19) を満たす表と、第 1 攪乱用データ $X1_i$ (1403) と、第 2 攪乱用データ $X2_i$ (1404) から、表変形手段 (1406) により (式 20) を満たす変換が行なわれ、変形済み表 (1407) を生成する。第 1 ハミングウェイト一定乱数生成手段 (1401) および第 2 攪乱用データ $X2_i$ (1404) には、請求項 6、請求項 7、請求項 8 のいずれかのハミングウェイト一定乱数生成手段を用いることが出来る。

【0041】

図 15 は、図 13 の実施例で用いる、第 1 攪乱用データ $X1_i$ (1505) および第 2 攪乱用データ $X2_i$ (1506) および変形済み表 (1509) を作成するための一実施例で、請求項 11 の一実施例である。予めハミングウェイト一定のデータが複数個格納された、第 1 攪乱用データ格納手段 (1501) と、第 1 攪乱用データ格納手段 (1501) から値を 1 つ選択して取り出し、第 1 攪乱

用データ $X1i$ (1505) とするための第1 攪乱用データ選択手段 (1503) と、第2 攪乱用データ格納手段 (1502) から値を1つ選択して取り出し、第2 攪乱用データ $X2i$ (1506) とするための第1 攪乱用データ選択手段 (1504) と、表格納手段 (1507) に格納された、(式19) を満たす表と、第1 攪乱用データ $X1i$ (1505) と、第2 攪乱用データ $X2i$ (1506) から、表変形手段 (1508) により (式20) を満たす変換が行なわれ、変形済み表 (1407) が生成される。

【0042】

図56に第1 攪乱用データ格納手段 (1501) に格納された第1 攪乱用データ、及び第2 攪乱用データ格納手段 (1502) に格納された第2 攪乱用データの例を示す。図57に、表格納手段 (1507) に格納された表の例を示す。この例は、第1 攪乱用データが $0x1c71c71c71c7$ 、第1 攪乱用データが $0x55555555$ が選択された例である。

【0043】

図16は、図13の実施例で用いる、第1 攪乱用データ $X1i$ (1603) および第2 攪乱用データ $X2i$ (1604) および変形済み表 (1605) を作成するための一実施例で、請求項12の一実施例である。予め第1 攪乱用データ $X1i$ に成り得るハミングウェイト一定の値と、第2 攪乱用データ $X2i$ に成り得るハミングウェイト一定のデータと、さらにそれらの組を用いて変形処理が行われた変形済み表が複数組が格納された、第1 攪乱用データ、第2 攪乱用データ及び変形済み表格納手段 (1602) から、第1 攪乱用データ、第2 攪乱用データ及び変形済み表選択手段 (1601) により、第1 攪乱用データ $X1i$ 、第2 攪乱用データ及び変形済み表の組が1つ選択して取り出され、第1 攪乱用データ $X1i$ (1603)、第2 攪乱用データ $X2i$ (1604)、変形済み表 (1605) として選択される。図58に図16に対応する第1 攪乱用データ、第2 攪乱用データ及び変形済み表のテーブルの例を示す。第1 攪乱用データ、第2 攪乱用データ及び変形済み表格納手段 (1602) には、このような表を複数セット有している。

【0044】

図 1 7 は、データ処理がテーブルルックアップとそれに引き続く 1 つの処理で定義された処理を、2 個の攪乱用データを用いて攪乱する実施例で、請求項 1 3 の一実施例である。入力データ D 1 (1 7 0 1) を用いて、表を引き、処理 p を行い、処理済データ D 2 (1 7 1 2) 得る。表を T a b l e とすると、D 1 と D 2 の関係は、(式 2 5) のようになる。

$$D 2 = p (T a b l e [D 1]) \quad (式 2 7)$$

この表引き処理を行う際の電流を測定すると、D 1 や D 2 の値が推定できる。そこで、第 1 攪乱用データ X 1 i と第 2 攪乱用データ X 2 i と、表のインデックスと出力結果を変形する関数 f と関数 g を考え、変形済みの表 X T a b l e を (式 2 7) のように定義する。

$$\begin{aligned} X T a b l e [f (i , X 1 i)] \\ = g (T a b l e [i] , X 2 i) \end{aligned} \quad (式 2 8)$$

また、g の逆関数を h と定義する。

$$D = h (g (D , X) , X) \quad (式 2 9)$$

X 2 o を次の様に定義する

$$X 2 o = p (X 2 i) \quad (式 3 0)$$

すると、表引き及び処理 p は、

$$H 1 = f (D 1 , X 1 i) \quad (式 3 1)$$

$$H 2 = X T a b l e [H 1] \quad (式 3 2)$$

$$H 3 = p (H 2) \quad (式 3 3)$$

$$D2 = h(H3, X2o) \quad (\text{式} 34)$$

と表現される。ここで、関数 f 、関数 h 、関数 p は、(式 35) を満たすものである必要がある。

$$a = h(p(f(a, X)), p(X)) \quad (\text{式} 35)$$

この様な条件を満たす関数 f 、関数 h 、関数 p の組み合わせには、

$$f(x, y) = x \text{ xor } y \quad (\text{式} 36)$$

$$p(x) = \text{右ローテーション}(x)$$

$$h(x, y) = x \text{ xor } y$$

などがある。

【0045】

ここで、(式 32) の処理を行なう際の消費電流により $H1$ が推定されても、 $X1i$ により攪乱されているため、(式 32) の測定結果だけでは、 $D1$ の値を推定することは出来ない。同様に、(式 33) の処理の消費電流により $H3$ が推定されても、 $X2i$ により攪乱されているため、(式 33) の測定結果だけでは、 $D2$ の値を推定することは出来ない。図 17 では、(式 31) にあたる処理は、第 1 攪乱用データ $X1i$ (1703) を用いて、入力データ $D1$ (1701) をデータ変形処理手段 (1702) で処理し、変形データ $H1$ (1704) を得る処理に相当する。また、(式 32) に当たる処理は、変形データ $H1$ (1704) をインデックスとして、変形済み表 (1706) を表引きする変形済み表アクセス手段 (1705) を用いて、変形データ $H2$ (1707) を得る処理に相当する。(式 33) に相当する処理は、変形データ $H2$ (1707) を変形済みデータ処理手段 (1708) により処理し、処理済変形データ $H3$ (1709) を得る処理に相当する。(式 34) に相当する処理は、処理済変形データ $H3$ (1709) を処理済第 2 攪乱用データ $X2o$ (1711) を用いてデータ逆変形

処理手段（1710）により逆変形し、処理済データD2（1712）を得る処理に相当する。

【0046】

図19は、図17の実施例で用いる、第1攪乱用データX1i（1903）および第2攪乱用データX2i（1904）および変形済み表（1908）および処理済第2攪乱用データX2o（1909）を作成するための一実施例で、請求項14の一実施例である。

【0047】

第1ハミングウェイト一定乱数生成手段（1901）により、第1攪乱用データX1i（1903）を生成し、第2ハミングウェイト一定乱数生成手段（1902）により、第2攪乱用データX2i（1904）が生成される。攪乱用データ処理手段（1907）により、第2攪乱用データX2i（1904）が処理され、処理済第2攪乱用データ（1909）が生成される。ハミングウェイト検査手段（1910）により、処理済第2攪乱用データ（1909）のハミングウェイトが検査され、不適切な場合は、第2ハミングウェイト一定乱数生成手段（1902）に対して、再生成制御信号が送られ、第2攪乱用データX2i（1904）の再生成が行われる。表格納手段（1905）に格納された表と、第1攪乱用データX1i（1903）と、第2攪乱用データX2i（1904）から、表変形手段（1906）により（式27）を満たす変換が行なわれ、変形済み表（1908）を生成する。第1ハミングウェイト一定乱数生成手段（1901）および第2攪乱用データX2i（1904）には、請求項6、請求項7、請求項8のいずれかのハミングウェイト一定乱数生成手段を用いることが出来る。本実施例の利点は、第1攪乱用データ、第2攪乱用データをその都度生成するため、特に攪乱用データのビット長が長い場合に、多くのバリエーションが期待できる点である。

【0048】

図20は、図17の実施例で用いる、第1攪乱用データX1i（2005）および第2攪乱用データX2i（2006）および変形済み表（2010）および処理済第2攪乱用データX2o（2011）を作成するための一実施例で、請求

項15の一実施例である。

【0049】

第1攪乱用データ格納手段(2001)に格納された、複数の第1攪乱用データ候補から、第1攪乱用データ選択手段(2003)によりデータを選択し、第1攪乱用データX1i(2005)を生成し、第2攪乱用データ格納手段(2002)に格納された、複数の第2攪乱用データ候補から、第2攪乱用データ選択手段(2004)によりデータを選択し、第2攪乱用データX2i(2006)を生成する。攪乱用データ処理手段(2009)により、第2攪乱用データX2i(2006)が処理され、処理済第2攪乱用データ(2011)が生成される。表格納手段(2007)に格納された表と、第1攪乱用データX1i(2005)と、第2攪乱用データX2i(2006)から、表変形手段(2008)により(式26)を満たす変換が行なわれ、変形済み表(2010)が生成される。本実施例の利点は、予め第1攪乱用データ、第2攪乱用データの項補が用意されているため、第1攪乱用データおよび第2攪乱用データの生成に時間がかからず、かつリーク情報も少なく済むことである。

【0050】

図21は、図17の実施例で用いる、第1攪乱用データX1i(2105)および第2攪乱用データX2i(2106)および変形済み表(2110)および処理済第2攪乱用データX2o(2107)を作成するための一実施例で、請求項16の一実施例である。

【0051】

第1攪乱用データ格納手段(2101)に格納された、複数の第1攪乱用データ候補から、第1攪乱用データ選択手段(2103)によりデータを選択し、第1攪乱用データX1i(2105)を生成し、第2攪乱用データおよび処理済第2攪乱用データ格納手段(2102)に格納された、複数の第2攪乱用データおよび処理済第2攪乱用データ候補の組から、第2攪乱用データおよび処理済第2攪乱用データ選択手段(2104)によりデータを選択し、第2攪乱用データX2i(2106)をおよび処理済第2攪乱用データX2o(2107)を生成する。表格納手段(2109)に格納された表と、第1攪乱用データX1i(21

05)と、第2攪乱用データX2i(2106)から、表変形手段(2108)により(式26)を満たす変換が行なわれ、変形済み表(2110)が生成される。本実施例の利点は、予め第1攪乱用データ、第2攪乱用データおよび処理済第2攪乱用データの項補が用意されているため、第1攪乱用データ、第2攪乱用データおよび処理済み第2攪乱用データX2oの生成に時間がかからず、第2攪乱用データから処理済第2攪乱用データを生成する処理がなく、その分請求項15に比べてさらにリーク情報が少なくて済む。

【0052】

図22は、図17の実施例で用いる、第1攪乱用データX1i(2203)および処理済第2攪乱用データX2i(2104)および変形済み表(2205)を作成するための一実施例で、請求項17の一実施例である。

【0053】

第1攪乱用データと処理済第2攪乱用データと変形済み表の組を複数格納した第1攪乱用データ、処理済第2攪乱用データおよび変換済み表格納手段(2202)より、第1攪乱用データ処理済み第2攪乱用データおよび変換済み表選択手段(2201)により、第1攪乱用データ処理済み第2攪乱用データおよび変換済み表を選択し、第1攪乱用データX1i(2203)および処理済第2攪乱用データX2i(2104)および変形済み表(2205)とする。本実施例の利点は、表を変形する処理が不要なため、請求項16に比べ、さらにリーク情報が少なくて済む。

【0054】

図23は、表引きとそれに引き続く1つの処理からなる一連の処理を複数回繰り返す処理装置において、表のインデックスと内容を2つの攪乱用データを用いて処理中に現れる数値を攪乱し、処理を行なう情報処理装置の1実施例で、請求項18の1実施例である。

【0055】

複数回繰り返して処理を行なう際に、変形されているデータを逆変形する際に、まず次の処理のための変形を行なった後に、逆変形を行なうという手順を取るため、変形されていないデータが途中では現れず、途中では、2重に変形されて

いるか、1重に変形されているかの違いはあるが、常に途中のデータは変形されたまま処理されるため、リーク情報が少ないのが特徴である。

【0056】

この請求項18で用いられるデータ変形およびデータ逆変形手段は、実行する順番を変えても同じ値を返す必要がある。データ x を攪乱情報 y 変形する処理を関数 $f(x, y)$ とし、変形データ x を攪乱情報 y で逆変形する関数を $g(x, y)$ とすると、

$$f(g(x, y_1), y_2) = g(f(x, y_2), y_1) \quad (\text{式37})$$

なる条件を満たす必要がある。

【0057】

実施例中の、第1攪乱用データ $X1i$ (2303)、変形済み表 (2306)、変形済み第2攪乱用データ (2313) は、請求項14、請求項15、請求項16、請求項17の実施例のいずれかにより生成することが可能である。入力データ $D1$ (2301) を第1攪乱用データ $X1i$ を用いてデータ変形処理手段 (2302) で変形し、変形データ $H1$ (2304) を得る。変形データ $H1$ (2304) を用いて、変形済み表 (2306) を変形済み表アクセス処理手段 (2305) により表引き処理を行い、変形データ $H2$ (2307) を得る。変形済みデータ処理手段 (2308) にて、変形データ $H2$ (2307) を処理し、処理済変形データ $H3$ (2309) を得る。この状態で、 $H2$ は、第2攪乱用データによる変形を受けた状態にある。さらに、第1攪乱用データ $X1i$ (2303) を用いて、処理済変形データ $H3$ (2309) を、データ変形処理手段 (2310) で変形し、処理済変形データ $H4$ (2311) を得る。 $H4$ は、第1攪乱用データと第2攪乱用データの2つにより変形された状態にある。ここで、変形済み第2攪乱用データ $X2o$ (2313) を用いて、データ逆変形処理手段 (2312) により、処理済変形データ $H5$ (2314) を得る。 $H5$ は、第2攪乱用データによる変形が解かれ、第1攪乱用データのみによって変形している状態となり、変形済み表のインデックスとして使用可能である。処理済変形データ H

5 (2314) を用いて、変形済み表アクセス処理手段 (2315) により、変形済み表 (2306) の表引きを行い、処理済変形データ H6 (2316) を得る。処理済変形データ H6 (2316) を変形済みデータ処理手段 (2317) で処理し、処理済変形データ H7 (2318) を得る。H7 は、第2攪乱データで変形を受けているため、最終結果とするために、変形済み第2攪乱データ X2o (2313) を用いて、逆変形を行い、最終的な結果である、処理済データ D2 (2320) を得る。この実施例は、一連の処理の繰り返し回数は、2回のみであるが、同様の手順で複数回の繰り返しが可能である。

【0058】

図19は、図23の実施例で用いる、第1攪乱用データ X1i (1903) および変形済み表 (1908) および処理済第2攪乱用データ X2o (1909) を作成するための一実施例でもあり、図19と図23をあわせた実施例は、請求項19の一実施例である。

【0059】

図20は、図23の実施例で用いる、第1攪乱用データ X1i (2005) および変形済み表 (2010) および処理済第2攪乱用データ X2o (2011) を作成するための一実施例でもあり、図20と図23をあわせた実施例は、請求項20の一実施例である。

【0060】

図21は、図23の実施例で用いる、第1攪乱用データ X1i (2105) および変形済み表 (2110) および処理済第2攪乱用データ X2o (2107) を作成するための一実施例でもあり、図21と図23をあわせた実施例は、請求項21の一実施例である。

【0061】

図22は、図23の実施例で用いる、第1攪乱用データ X1i (2203) および変形済み表 (2204) および処理済第2攪乱用データ X2o (2205) を作成するための一実施例でもあり、図22と図23をあわせた実施例は、請求項22の一実施例である。

【0062】

図 2 4 は、表引きとそれに引き続く 1 つの処理からなる一連の処理を複数回繰り返す処理装置において、表のインデックスと出力をそれぞれ 2 つの攪乱用データで 2 重に変形し、処理する実施例の一つで、請求項 2 3 の実施例の一つである。2 重に変形することにより、消費電流による観測を少ない資源でより強力な攪乱が行なえるようにする。

【 0 0 6 3 】

この実施例で用いられる、4 つの攪乱用データと、前記攪乱用データを用いて第 2 変形済み表を生成する方法としては、請求項 1 9、2 0、2 1、2 2 がある。たとえば、予めハミングウェイト一定の数値を複数個用意して選択する実施例の場合、予め用意した数値のパリエーションが少ない場合、攪乱データを用いてデータを変形する処理が既知の場合、予め用意されている攪乱用データを全て推定することも可能となる。攪乱用データを用いて変形する処理が XOR 処理であった場合、元のデータと攪乱用データが同一のデータであった場合、変形済みのデータは 0 となることから、予め用意された攪乱用データのセットを推定することが不可能ではない。そこで、たとえばハミングウェイトを固定しない代りに常にビット長で表現できる全ての値を生成した攪乱用データで変形した後に、ハミングウェイト一定の攪乱用データを用いてさらに変形を行なうことで、容易には攪乱用データの推定ができなくなる。図 2 4 を用いて、処理の詳細を説明する。

【 0 0 6 4 】

入力データ D 1 (2 4 0 1) が、第 3 攪乱用データ X 3 i (2 4 0 3) を用いて、データ変形手段 (2 4 0 2) で変形され、変形データ H 1 (2 4 0 4) が生成される。第 3 攪乱用データは、表引き処理に用いられる表のインデックスを変形する 2 つの攪乱データのうちのひとつで、実際の表引きには、さらに第 1 攪乱用データ X 1 i (2 4 0 6) で変形する必要がある。その変形処理を行なうのが、データ変形処理手段 (2 4 0 5) で、変形データ H 1 (2 4 0 4) を第 1 攪乱用データ (2 4 0 6) を用いて変形し、変形データ H 2 (2 4 0 7) を得る。得られた変形データ H 2 (2 4 0 7) をインデックスとして用い、変形済み表アクセス処理手段 (2 4 0 8) により第 2 変形済み表 (2 4 0 9) を表引きし、変形データ H 3 (2 4 1 0) を得る。変形データ H 3 (2 4 1 0) を処理する変形済み

データ処理手段（2411）で処理し、処理済変形データH4（2412）を生成する。第3攪乱用データX3i（2403）を用いて、データ変形処理手段（2413）により、処理済変形データH4（2412）を変形し、処理変形データH5（2414）を得て、さらに第1攪乱用データX1i（2406）を用いてデータ変形処理手段（2415）で変形し、処理済変形データH6（2416）を得る。ここまでの、処理中のデータは、第3攪乱用データX3i（2403）、第1攪乱用データX1i（2406）、および第2変形表（2409）に畳み込まれている、処理済第2攪乱用データX2o（2418）、処理済第4攪乱用データX4o（2421）で変形されている状態なので、まずデータ逆変換手順（2417）で、処理済第2攪乱用データX2o（2418）を用いて、データの逆変形を行い、処理済変形データH7（2419）を生成する。そしてデータ逆変換手順（2420）で、処理済第4攪乱用データX4o（2421）を用いて、データの逆変形を行い、処理済変形データH8（2422）を生成する。これで、データは第3攪乱用データX3i（2403）、第1攪乱用データX1i（2406）の2つの攪乱用データで変形されている状態になるので、第2変形済み表（2409）のインデックスとして用いることが出来る。変形済み表アクセス処理手順（2423）で表引きを行い、変形データH9（2424）を得て、さらに変形済みデータ処理手順（2425）により、処理済変形データH10（2426）を得る。ここまでの、最終結果を処理済第2攪乱用データX2o（2418）、処理済第4攪乱用データX4o（2421）で変形されている状態のデータが得られたことになるので、データ逆変換処理手順（2427）により、処理済第2攪乱用データX2o（2418）による変形分を逆変形し、さらにデータ逆変換処理手順（2429）により、処理済第4攪乱用データX4o（2421）による変形分を逆変形して最終的な結果D2（2530）を得る。

【0065】

図26は、図24すなわち請求項23の実施例における、第1攪乱用データX1i（2602）、処理済第2攪乱用データX2o（2606）、第3攪乱用データX3i（2612）、処理済第4攪乱用データX4o（2618）と、第2処理済表（2617）を生成する方法の1実施例であり、請求項24の一実施例

である。処理手順は、まず図 1 9 の実施例の方法で、第 1 攪乱用データ X_{1i} (2602)、処理済第 2 攪乱用データ X_{2o} (2606)、および変形済み表 (2610) を生成し、生成した変形済み表 (2610) をさらに第 3 攪乱用データ生成手段 (2611) で生成された第 3 攪乱用データ X_{3i} (2612) と、第 4 攪乱用データ生成手段 (2613) で生成された第 4 攪乱用データ X_{4i} (2614) とを用いて、表変形処理手段 (2616) にて変形済み表 (2610) をさらに変形して、第 2 変形済み表 (2617) を得る。また、データの逆変換時に必要となる、処理済み第 4 攪乱用データ X_{4o} (2618) を第 4 攪乱用データ X_{4i} (2614) から、攪乱用データ処理手段 (2615) を用いて計算しておく。

【0066】

図 2 7 は、図 2 4 すなわち請求項 2 3 の実施例における、第 1 攪乱用データ X_{1i} (2703)、処理済第 2 攪乱用データ X_{2o} (2707)、第 3 攪乱用データ X_{3i} (2712)、処理済第 4 攪乱用データ X_{4o} (2718) と、第 2 処理済表 (2714) を生成する別の方法の位置実施例であり請求項 2 5 の一実施例である。

【0067】

処理手順は、まず図 2 0 の実施例の方法で、第 1 攪乱用データ X_{1i} (2703)、処理済第 2 攪乱用データ X_{2o} (2706)、および変形済み表 (2710) を生成し、生成した変形済み表 (2710) をさらに第 3 攪乱用データ生成手段 (2711) で生成された第 3 攪乱用データ X_{3i} (2712) と、第 4 攪乱用データ生成手段 (2715) で生成された第 4 攪乱用データ X_{4i} (2716) とを用いて、表変形処理手段 (2713) にて変形済み表 (2710) をさらに変形して、第 2 変形済み表 (2714) を得る。また、データの逆変換時に必要となる、処理済み第 4 攪乱用データ X_{4o} (2718) を第 4 攪乱用データ X_{4i} (2716) より、攪乱用データ処理手段 (2717) を用いて計算しておく。

【0068】

図 2 8 は、図 2 4 すなわち請求項 2 3 の実施例における、第 1 攪乱用データ X

1 i (2904)、処理済第2攪乱用データX2o (2805)、第3攪乱用データX3i (2807)、処理済第4攪乱用データX4o (2804)と、第2処理済表(2909)を生成する別の方法の位置実施例であり請求項26の一実施例である。

【0069】

処理手順は、まず図21の実施例の方法で、第1攪乱用データX1i (2804)、処理済第2攪乱用データX2o (2805)、および変形済み表(2803)を生成し、生成した変形済み表(2803)をさらに第3攪乱用データ生成手段(806)で生成された第3攪乱用データX3i (2807)と、第4攪乱用データ生成手段(2810)で生成された第4攪乱用データX4i (2811)とを用いて、表変形処理手段(2713)にて変形済み表(2803)をさらに変形して、第2変形済み表(2809)を得る。また、データの逆変換時に必要となる、処理済第4攪乱用データX4o (2813)を第4攪乱用データX4i (2811)より、攪乱用データ処理手段(2812)を用いて計算しておく。

【0070】

図29、30、31、32、33、34、35、36、37、38、39、45、46、47を用いて、請求項27の一実施例について説明する。

【0071】

まず、図29を用いて、SBOX表の変形処理および攪乱用データについて説明する。SBOX表のアドレス攪乱用データSinX1 (2901)とSBOXデータ攪乱用データSoutX (2902)を用いて、SBOX表(2903)をSBOX表変形手段(2904)で変形処理を行い、変形済みSBOX表(2905)を得る。SBOX表のアドレスおよびデータの変形は、XORで行なうものとする。また、SoutXを転置処理P (2906)、転置処理E (2907)を行い、転置済みSBOXデータ攪乱用データXSoutX (2909)を生成しておく。ここでの処理を纏めると、SBOX表をSBOX[0..63]、変形SBOX表をXSBOX[0..63]、転置処理Pを関数P ()、拡大転置処理Eを関数E ()で表現すると、

$$X S B O X [i \text{ xor } SinX1] = SB0X[i] \text{ xor } SoutX \quad (\text{式 } 38)$$

$$X S o u t X = E (P (S o u t X)) \quad (\text{式 } 39)$$

となる。SinX1、SoutXの作成方法は、請求項14、15、16、17のいずれかの方法を用いることができる。

【0072】

図46は、SB0Xの格納様式の一実施例である。64個の32ビット整数からなる要素を持つ1次元の配列としてSB0Xを格納する。図45は、図46の様式で格納されたSB0Xを表引きするための処理の一実施例で、48ビットの入力を6ビットごとに分解し、SB0X配列を表引きし、表引きの際に用いた6ビットの値の位置に応じたマスクで必要なデータを取り出し、順次加算することで、最終的な表引き結果を得る。図45のフローチャートを用いて、処理を順を追って説明する。4502で、inに表引きの入力となる48ビットの数値を代入する。48ビットを6ビットづつに分解し、8回に分けて処理するため、4503において、処理回数をカウントするための変数jを0に初期化する。4504で、表引き結果をマスクするための変数maskを15で初期化する。これは、下位4ビットがすべて1である数値である。4505で、表引き結果を格納する変数resultを0に初期化する。4506で、inの下位6ビットを取り出し、idxに代入する。4507で、inを右に6ビットシフトし、つぎの6ビットを取り出す準備を行う。4508でdに図46のSB0Xをidxをインデックスとしたときの表引き結果を代入する。4509でdとmaskのANDを計算し、dに代入する。4510で、resultにdを加える。4511にて、maskを左に4ビットシフトし、次のデータをマスクする準備とする。4512でカウンタjを1だけ加算する。4513でjが8未満であれば、4506からの処理を繰り返す。4514では、resultを結果として返す。

【0073】

図47は、図29のSB0X表変形手段(2304)の詳細なフローチャートであり、図46の様式で格納されたSB0Xを変形するための手順を示す。この

手順で変形された変形済み SBOX 表は、図 4 6 の表引き処理により、表引きが可能である。また、図 4 7 で選られた変形済み SBOX を通常の SBOX と見なし、新たな攪乱用データを用いて図 4 7 の処理を繰り返すことで、多重に変形することが可能である。図 4 7 にしたがって、SBOX 変形の手順を説明する。

4 7 0 2 では、SBOX を表引きするためのインデックス idx を 0 に初期化する。4 7 0 3 では、 idx を 6 ビットのビット列とみなし、そのビット列を 8 個繰り返して選られる 48 ビットの数値を計算し、 in に代入する。4 7 0 3 では、 in と、48 ビットアドレス攪乱用データとの xor を計算し、 in に代入する。4 7 0 5 では、 in を SBOX の表引き用の 48 ビットの入力として、図 4 5 に示される SBOX 表アクセス手順を呼び出し、結果を $result$ とする。4 7 0 6 では、結果 $result$ に対して、32 ビットの出力攪乱用データとの xor を計算し、 $result$ に代入する。4 7 0 7 では、変形済み SBOX 表のインデックス idx の位置に、 $result$ を代入する。4 7 0 8 では、 idx に 1 を加える。4 8 0 9 では、 idx が 64 未満かどうかを判定し、64 未満の場合は、4 7 0 3 からの処理を繰り返す。 idx が 64 になると、表の変換が終了する。

【 0 0 7 4 】

次に、図 3 0 を用いて、平文攪乱用データ PX で変形したデータを、逆変換するために用いる、置換済み平文攪乱用データ $PX \circ 1$ (3003)、 $PX \circ 2$ (3007)、 $PX \circ 3$ (3006)、 $PX \circ 4$ (3010) の生成について説明する。平文用データ PX を転置処理 IP (3002) で転置し、上位 32 ビットと下位 32 ビットにわけ、それぞれ転置済み平文攪乱用データ 1 の $PX \circ 1$ (3003)、転置済み平文攪乱用データ 2 の $PX \circ 2$ (3007) とする。この 2 つの値は、最終ラウンドの処理が終わった後の、 IP 逆転置処理の直前で変形データを最終的な結果にするための逆変形処理に用いられる。つぎに、転置済み平文攪乱用データ 1 の $PX \circ 1$ (3003)、転置済み平文攪乱用データ 2 の $PX \circ 2$ (3007) に対して、拡大転置処理 E (3005、3009) を行ない、転置済み平文攪乱用データ 3 の $PX \circ 3$ (3006)、転置済み平文攪乱用データ 4 の $PX \circ 4$ (3010) とし、各ラウンドでの変形 SBOX 表の表引き前の逆変形処理に

用いる。

【0075】

次に、図31を用いて、秘密鍵攪乱用データと、各ラウンドごとの鍵の処理のうち、LS処理の直後に行なう変形処理用に用いる、処理済み秘密鍵攪乱用データKX₀₁(3109)、KX₀₂(3111)、KX₀₃(3113)について、説明する。本実施例では、選択転置処理PC2での通常の出力を、Xとした場合、

$$X \text{ xor } X_{\text{Sout}} X \quad (\text{式40})$$

という値を出力するようにしたい。選択置換PC1をPC1()と表現し、LS処理をLS()と表現し、鍵をKと表現する。本実施例では、鍵は秘密鍵攪乱用データKXとのxorで攪乱されるため、第1ラウンドでは、

$$K_0 = \text{LS}(\text{PC1}(KX \text{ xor } K)) \quad (\text{式41})$$

$$KX_{01} = \text{LS}(\text{PC1}(KX)) \text{ xor } \text{INV_PC2}(\text{SinX1}) \quad (\text{式42})$$

$$K_1 = K_0 \text{ xor } KX_{01} \quad (\text{式43})$$

$$K_{1_OUT} = \text{PC2}(K_1) \quad (\text{式44})$$

K_{1_OUT}をPC2からの出力として使うことで、(式40)を満たす値が得られる。つぎに、第2ラウンド用の値は、

$$KX_{02} = \text{INV_PC2}(\text{SinX1}) \text{ xor } \text{LS}(\text{INV_PC2}(\text{SinX1})) \quad (\text{式45})$$

$$K_2 = \text{LS}(K_1) \text{ xor } KX_{02} \quad (\text{式46})$$

$$K2_OUT = PC2(K2)$$

となる。また、たとえばラウンド3のように、LS処理で2ビット分ローテートするラウンドでは、

$$KXo3 = INV_PC2(SinX1) \text{ xor } LS(LS(INV_PC2(SinX1))) \quad (\text{式47})$$

$$K3 = LS(LS(K2)) \text{ xor } KXo3$$

$$K3_OUT = PC2(K3) \quad (\text{式48})$$

とすることで、(式40)を満たす値が得られる。LS処理でシフトするビットの種類は2種類のみなので、ラウンド1用のKXo1と、LS処理のシフト量が1ビットのKXo2、LS処理のシフト量が2ビットのKXo3の3種類の値が必要となる。これらの3つの値で、ラウンド16全ての変形処理を行なうことが出来る。(式41)、(式45)、(式47)で示されるKXo1、KXo2、KXo3の計算は、図31のシグナルフローの通りである。本実施例では、変形処理にxor演算を用いるので、図31中の、逆変換結合処理(3108、3110、3112)はxor処理を用いる。

【0076】

図32は、平文データPtext(3201)を変形する処理の一実施例である。

【0077】

平文攪乱用データPX(3203)を用いて、第1変形処理(3202)で変形し、変形済み平文データXPtext(3204)を作成する。ここで、第1変形処理(3202)は、本実施例では、xor演算を用いる。したがって、

$$XPtext = Ptext \text{ xor } PX \quad (\text{式49})$$

となる。XPtextを転置処理IP(3205)により転置し、上位32ビット

トを転置済み変形平文1のX P t e x t L (3 2 0 6)、下位32ビットを転置済み変形平文2のX P t e x t R (3 2 0 7)とする。この処理は、第1変形処理(3 2 0 2)を除けば、通常のDES暗号処理の処理フローに等しい。

【 0 0 7 8 】

ここで、別の実施例を図54に示す。この別の実施例では、平文データ(5 4 0 1)を転置処理I P (5 4 0 2)で処理した後に、第1変形手段(5 4 0 4、5 4 0 7)で攪乱している。この実施例では、平文攪乱用データP X (5 3 0 1)の処理が、転置処理I Pを行わずに済む分だけ、効率的である。図53は、平文攪乱用データの処理の1実施例である。

【 0 0 7 9 】

図33は、秘密鍵データK (3 3 0 1)の変形処理の一実施例で、秘密鍵攪乱用データK X (3 3 0 3)を用いて、第2変形処理(3 3 0 2)にて、秘密鍵データK (3 3 0 1)を変形し、変形済み秘密鍵データX K (3 3 0 4)を得る。本実施例では、第2変形処理手段として、x o r演算を用いる。したがって、

$$X K = K \quad x o r \quad K X \quad (\text{式} 5 0)$$

となる。つぎに各ラウンドの処理を図34、図35、図36、図37、図38に示す。

【 0 0 8 0 】

5つの図の違いは、ラウンドの違いで、第1ラウンド、第5ラウンド、第9ラウンド、第13ラウンドの処理が図35、第2ラウンド、第6ラウンド、第10ラウンド、第14ラウンドの処理が、図35、第3ラウンド、第7ラウンド、第11ラウンド、第15ラウンドの処理が、図36、第4ラウンド、第8ラウンド、第12ラウンドの処理が図37、第16ラウンドの処理が図38に示されている。

【 0 0 8 1 】

図34の処理を順を追って説明する。まず、攪乱処理を行わないときのX P t e x t L (3 4 0 1)に来るべき値をP t e x t L、同様に攪乱処理を行わな

ったときのX P t e x t R (3402) に来るべき値をP t e x t Lとする。すると、X P t e x t LおよびX P t e x t Rは、

$$X P t e x t L = P T e x t L \text{ xor } P X o 1 \quad (\text{式51})$$

$$X P t e x t R = P T e x t R \text{ xor } P X o 2 \quad (\text{式52})$$

と表現される。同様に、X K L (3407) の値をX K L 0、攪乱処理を行わない場合のX K L (3407) に来るべき値をK L、選択置換処理P C 1を関数P C 1 () として表現すると、

$$X K L 0 = K L \text{ xor } P C 1 (K X) \quad (\text{式53})$$

となる。第3変形処理(3409)を実行後のX K L (3410)の値をX K L 1とし、選択置換P C 2の逆関数をI N V _ P C 2 () とする。ここで、P C 2 () で参照されないビットに付いては、I N V _ P C 2 () では0になるものとする。第3逆変形処理(3409)で使用される処理済秘密鍵攪乱用データは、そのラウンドで行われるローテート処理L S (3408)でシフトされるビット数で決まり、1ビットの場合は、K X o 2、2ビットの場合は、K X o 3、ラウンド1の場合は、K X o 1が用いられる。

$$X K L 1 = L S (X K L 0) \text{ xor } K X o 1 \quad (\text{式54})$$

さらに、(式53)と(式42)を代入すると、

$$\begin{aligned} X K L 1 \\ = L S (K L \text{ xor } P C 1 (K X)) \text{ xor } K X o 1 \quad (\text{式55}) \end{aligned}$$

また、

$$LS(a \text{ xor } b) = LS(a) \text{ xor } LS(b) \quad (\text{式56})$$

$$\begin{aligned} & (a \text{ xor } b) \text{ xor } c \\ &= a \text{ xor } (b \text{ xor } c) \end{aligned} \quad (\text{式57})$$

という関係を用いて、(式54)を書き直すと、

$$\begin{aligned} XKL1 &= LS(KL \text{ xor } PC1(KX)) \text{ xor} \\ & (LS(PC1(KX)) \text{ xor} \\ & INV_PC2(SinX1)) \\ &= LS(KL \text{ xor } PC1(KX) \text{ xor} \\ & PC1(KX)) \text{ xor } INV_PC2(SinX1) \\ &= LS(KL) \text{ xor } INV_PC2(SinX1) \end{aligned} \quad (\text{式58})$$

となる。この値に選択置換処理PC-2(3414)を施した値をXKL1PC2とすると、

$$\begin{aligned} XKL1PC2 &= PC2(XKL1) \\ &= PC2(LS(KL) \text{ xor } INV_PC2(SinX1)) \\ &= PC2(LS(KL)) \text{ xor } SinX1 \end{aligned} \quad (\text{式59})$$

となる。ここでは、ラウンド1の場合について説明したが、ラウンド5、ラウンド9、ラウンド13においても、選択置換処理PC-2(3414)の出力は、攪乱処理を行わない場合の値である PC2(LS(KL))とSinX1のxorを行った値となる。

【0082】

つぎにxor演算(3404)を行った結果を、XPtextRXとし、拡大転置処理E(3403)の演算を関数E()と表現すると、XPtextRXは

$$XPtextRX$$

$$=E(XPtextR) \text{ xor } XKLP C2 \quad (\text{式60})$$

となり、(式52)、(式59)を代入すると、

$$\begin{aligned} XPtextRX &= E(PtextR \text{ xor } PXo2) \text{ xor} \\ &PC2(LS(KL)) \text{ xor } SinX1 \\ &= E(PtextR) \text{ xor } PC2(LS(KL)) \text{ xor} \\ &E(PXo2) \text{ xor } SinX1 \quad (\text{式61}) \end{aligned}$$

となる。転置済み平文攪乱用データ4の $PXo4$ (3416)を用いて、第1逆変形処理(3415)で逆変形した結果を、 $XPtextRX2$ とすると、第1逆変形処理はxor演算であるので、

$$\begin{aligned} XPtextRX2 &= XPtextR \text{ xor } PXo4 \\ &= E(PtextR) \text{ xor } PC2(LS(KL)) \\ &\text{ xor } E(PXo2) \text{ xor } SinX1 \\ &\text{ xor } PXo4 \quad (\text{式62}) \end{aligned}$$

また、図30の実施例より、

$$PXo4 = E(PXo2) \quad (\text{式63})$$

であるので、(式62)は、

$$\begin{aligned} XPtextRX2 &= E(PtextR) \text{ xor } PC2(LS(KL)) \\ &\text{ xor } E(PXo2) \text{ xor } SinX1 \\ &\text{ xor } E(PXo2) \\ &= E(PtextR) \text{ xor } PC2(LS(KL)) \end{aligned}$$

$$\text{xor SinX1} \quad (\text{式64})$$

となる。また、攪乱処理を行わない場合の、変形SBOXアクセス処理(3418)の入力となる値PtextRX2は、

$$\begin{aligned} & \text{PtextRX2} \\ & = E(\text{PtextR}) \quad \text{xor} \quad \text{PC2}(\text{LS}(\text{KL})) \quad (\text{式65}) \end{aligned}$$

であり、(式64)は、

$$\begin{aligned} & \text{XPtextRX2} \\ & = \text{PtextRX2} \quad \text{xor} \quad \text{SinX1} \quad (\text{式66}) \end{aligned}$$

となる。攪乱処理を行わない値と比較すると、SBOXのアドレス攪乱用データSinX1とxorを取った値と等しくなるため、変形済みSBOX表(3419)をアクセス屢事ができる。アクセスした結果には、SoutXによる変形が行われているので、xor処理(3421)への入力は、攪乱を行わない場合のSBOXの出力をSResultとすると、

$$P(\text{SResult} \quad \text{xor} \quad \text{SoutX}) \quad (\text{式67})$$

となる。(式67)と転置済み変形平文1のXPtextL(3401)とのxor(3421)を計算すると、

$$\begin{aligned} & P(\text{Sresult} \quad \text{xor} \quad \text{SoutX}) \quad \text{xor} \quad \text{XPtextL} \\ & = P(\text{Sresult}) \quad \text{xor} \quad P(\text{SoutX}) \quad \text{xor} \quad \text{PtextL} \quad \text{xor} \\ & \quad \text{PTextL} \quad \text{xor} \quad \text{PXol} \quad (\text{式68}) \end{aligned}$$

となる。この値は、転置済み変形平文2のXPtextR(3423)に代入されるが、ここで、攪乱を行わない場合のXPtextR(3423)に代入される値を、PtextR2、攪乱を行った場合の値をXPtextR2とすると、

$$\begin{aligned} \text{XPtextR2} &= \text{PtextR2} \text{ xor } \text{P(SoutX)} \\ &\text{xor PXo1} \end{aligned} \quad (\text{式69})$$

同様にして、攪乱を行わない場合のXPtextL(3422)に代入される値を、PtextL2、攪乱を行った場合の値をXPtextL2とすると、

$$\text{XPtextL2} = \text{PtextL2} \text{ xor } \text{PXo2} \quad (\text{式70})$$

(式69)、(式70)の値は、次のラウンドの図35で使用される。(式69)と(式51)を比較した場合、PXo2の代わりにPXo1が使われ、さらにP(SoutX)でのxorが加わっている。この違いにより、図34と図35では、第1逆変形処理(3515)に用いられる転置済み平文攪乱用データ(3516)が、PXo4からPXo3に替わり、P(SoutX)による変形を元に戻すために、第4逆変形処理(3517)の処理が加わっている。第4逆変形処理(3517)にたどり着くまでに、P(SoutX)は、拡大置換処理E()が施され、E(P(SoutX))となり、転置済みSBOX攪乱用データXSoutXと等しくなる。

【0083】

図35の攪乱を行わない場合のXPtextR(3525)に代入される値を、PtextR3、攪乱を行った場合の値をXPtextR3とし、攪乱を行わない場合のXPtextL(3524)に代入される値を、PtextL3、攪乱を行った場合の値をXPtextL3とすると、

$$\begin{aligned} \text{XPtextR3} &= \text{PtextR3} \text{ xor } \text{P(SoutX)} \\ &\text{xor PXo2} \end{aligned} \quad (\text{式71})$$

$$\begin{aligned} \text{XPtextL3} &= \text{PtextL3} \quad \text{xor} \quad \text{P}(\text{SoutX}) \\ \text{xor} \quad \text{PXo1} & \quad \quad \quad (\text{式72}) \end{aligned}$$

となる。

【0084】

(式71)、(式72)の値は、次のラウンドの図36で使用される。(式71)と(式69)を比較した場合、PXo1の代わりにPXo2が使われている。この違いにより、図35と図36では、第1逆変形処理(3615)に用いられる転置済み平文攪乱用データ(3516)が、PXo3からPXo4に替わっている。また、xor処理(3623)の2つの入力のいずれにも、P(SoutX)がxorされているため、P(SoutX)の影響が消える。その結果、図36の攪乱を行わない場合のXPtextR(3625)に代入される値を、PtextR4、攪乱を行った場合の値をXPtextR4とし、攪乱を行わない場合のXPtextL(3624)に代入される値を、PtextL4、攪乱を行った場合の値をXPtextL4とすると、

$$\text{XPtextR4} = \text{PtextR4} \quad \text{xor} \quad \text{PXo1} \quad (\text{式73})$$

$$\begin{aligned} \text{XPtextL4} &= \text{PtextL4} \quad \text{xor} \quad \text{P}(\text{SoutX}) \\ \text{xor} \quad \text{PXo2} & \quad \quad \quad (\text{式74}) \end{aligned}$$

となる。

【0085】

(式73)、(式74)の値は、次のラウンドの図37で使用される。(式73)と(式71)を比較した場合、PXo2の代わりにPXo1が使われており、またP(SoutX)によるxorが含まれていない。この違いにより、図36と図37では、第1逆変形処理(3715)に用いられる転置済み平文攪乱用データ(3516)が、PXo4からPXo3に替わっている。P(SoutX)による変形を元に戻す必要が無いため、第4逆変形処理が無くなっている。また、xor処理(3721)の2つの入力のいずれにも、P(SoutX)がx

orされているため、 $P(SoutX)$ の影響が消える。その結果、図37の攪乱を行わない場合の $XPtextR(3722)$ に代入される値を、 $PtextR5$ 、攪乱を行った場合の値を $XPtextR5$ とし、攪乱を行わない場合の $XPtextL(3724)$ に代入される値を、 $PtextL5$ 、攪乱を行った場合の値を $XPtextL5$ とすると、

$$XPtextR5 = PtextR5 \text{ xor } PXo2 \quad (\text{式75})$$

$$XPtextL5 = PtextL5 \text{ xor } PXo1 \quad (\text{式76})$$

となり、(式75)、(式76)は(式51)、(式52)と同様の変形が行われているため、次のラウンドの処理が図34の実施例で行うことが出来る。

【0086】

図38は、図37の処理とほぼ同様であるが、最後に $XPtextL$ と $XPtextR$ の交換を行っていない。したがって、図38の攪乱を行わない場合の $XPtextR(3822)$ に代入される値を、 $PtextR6$ 、攪乱を行った場合の値を $XPtextR6$ とし、攪乱を行わない場合の $XPtextL(3824)$ に代入される値を、 $PtextL6$ 、攪乱を行った場合の値を $XPtextL6$ とすると、

$$XPtextR6 = PtextR6 \text{ xor } PXo1 \quad (\text{式77})$$

$$XPtextL6 = PtextL6 \text{ xor } PXo2 \quad (\text{式78})$$

となる。

【0087】

図39は、最終結果を求めるための処理フローを示す。第5逆変換処理(3905)により、転置済み平文攪乱用データ1の $PXo2(3904)$ を用いて、転置済み変形平文1の $XPtextL(3901)$ を逆変形し、第6逆変換処理(3906)により、転置済み平文攪乱用データ2の $PXo1(3903)$ を用いて、転置済み変形平文2の $XPtextR(3902)$ を逆変形する。これにより、変形がすべて取り除かれる。

$$P_{textR6} = X_{P_{textR6}} \text{ xor } P_{X01} \quad (\text{式79})$$

$$P_{textL6} = X_{P_{textL6}} \text{ xor } P_{X02} \quad (\text{式80})$$

最後に、転置処理IP-1(3907)を用いて、第5逆変換処理(3905)の結果と第6逆変換処理(3906)の結果を入力として、転置処理を行い、最終的な暗号文データCtext(3908)を得る。最終的な暗号データCtextを得る直前まで、どの時点をとっても、データは変形を受けたままとなるので、電流波形から本来のデータを推測することが困難に成る。

【0088】

SBOX表アドレス攪乱用データSinX、SBOX表データ攪乱用データSoutXおよび変形SBOX表の作成を、図19もしくは図20もしくは図21もしくは図22のような実施例で行ない、ハミングウエイトが常に一定とし、電流波形から本来のデータを推測することがさらに困難となった実施例が、請求項28の実施例となる。

【0089】

請求項29の一実施例を、図39、40、41、42、43、44、45、46、47、52に示す。基本的な手順は請求項27の実施例と同じであるが、SBOXの変形を2回行っている点が、実施例27と違う。SBOXの変形の一実施例を図52に示す。攪乱用データとしては、SBOXアドレス攪乱用データSinX1(5201)、転置済みSBOXデータ攪乱用データXSout1(5210)、SBOXアドレス攪乱用データSinX2(5212)、転置済みSBOXデータ攪乱用データXSoutX2(5218)を用い、第2変形済みSBOX表(5214)を生成する。SBOX攪乱用データの個数が増えたため、図40に示される、ラウンド1、ラウンド5、ラウンド9、ラウンド13の処理の一実施例において、SBOXアドレス攪乱用データSinX2(4015)を用いて、第3変形処理(4014)を行う処理が追加となり、図41に示される、ラウンド2、ラウンド6、ラウンド10、ラウンド14の処理の一実施例において、SBOXアドレス攪乱用データSinX2(4115)を用いて、第3変形処理(4114)を行う処理と、転置済みSBOXデータ攪乱用データXSout

ut X 2 (4 1 2 1) を用いて、第 4 逆変形成理 (4 1 2 0) が追加となり、図 4 2 に示される、ラウンド 3、ラウンド 7、ラウンド 1 1、ラウンド 1 5 の処理の一実施例において、S B O X アドレス攪乱用データ S i n X 2 (4 2 1 5) を用いて、第 3 変形成理 (4 2 1 4) を行う処理と、転置済み S B O X データ攪乱用データ X S o u t X 2 (4 2 2 1) を用いて、第 4 逆変形成理 (4 2 2 0) が追加となり、図 4 3 に示される、ラウンド 4、ラウンド 8、ラウンド 1 2 の処理の一実施例において、S B O X アドレス攪乱用データ S i n X 2 (4 3 1 5) を用いて、第 3 変形成理 (4 3 1 4) を行う処理が追加となり、図 4 4 に示される、ラウンド 1 6 の処理の一実施例において、S B O X アドレス攪乱用データ S i n X 2 (4 4 1 5) を用いて、第 3 変形成理 (4 4 1 4) を行う処理が追加となっている。

【 0 0 9 0 】

S B O X 表アドレス攪乱用データ S i n X 1、S B O X 表アドレス攪乱用データ S i n X 2、S B O X 表データ攪乱用データ S o u t X 1、S B O X 表データ攪乱用データ S o u t X 2 及び第 2 変形 S B O X 表の作成を、図 2 6 もしくは図 2 7 もしくは図 2 8 のような実施例で行った場合が、請求項 3 0 の実施例となる。

【 0 0 9 1 】

S B O X 表アドレス攪乱用データ S i n X 1、S B O X 表アドレス攪乱用データ S i n X 2、S B O X 表データ攪乱用データ S o u t X 1、S B O X 表データ攪乱用データ S o u t X 2 及び第 2 変形 S B O X 表の作成を、図 2 6 もしくは図 2 7 もしくは図 2 8 のような実施例で行ない、かつハミングウエイトの検査を、全ビット通じてのハミングウエイトではなく、中央演算処理装置で一度に処理できるビット数に区切って評価した際にも、ハミングウエイトのが一定となるようにハミングウエイトの検査を実施したものが、請求項 3 1 の実施例となる。

【 0 0 9 2 】

【発明の効果】

本項発明によれば、I C カードチップでの処理データを変形することと、その変形に用いる攪乱用データの生成時に制約を加えることにより、消費電流の波形

から、処理や暗号鍵の推測を困難にする。

【図面の簡単な説明】

【図 1】

図 1 は、I C カードのハードウェア構成の例を示す図である。

【図 2】

図 2 は、I C カード用チップ内のハードウェア構成の例を示す図である。

【図 3】

図 3 は、消費電流の波形例を示す図である。

【図 4】

図 4 は、一つの攪乱用データを使ったデータ変形の手順の例を示すフローチャートである。

【図 5】

図 5 は、攪乱用データをあらかじめ複数用意し、選択することで攪乱用データを生成する、一つの攪乱用データを使ったデータ変形の手順の例を示すフローチャートである。

【図 6】

図 6 は、ハミングウェイト一定の乱数生成方法を示すフローチャートである。

【図 7】

図 7 は、ハミングウェイト一定の乱数生成方法の例を示すフローチャートである。

【図 8】

図 8 は、ハミングウェイト一定の値の一覧表作成方法の例を示すフローチャートである。

【図 9】

図 9 は、ハミングウェイトが一定でかつ、データ処理手順処理後のハミングウェイト一定の値の一覧表作成方法の例を示すフローチャートである。

【図 1 0】

図 1 0 は、ビット長の短いハミングウェイト一定の一覧表から、ビット長の長いハミングウェイト一定の乱数の生成方法の例を示すフローチャートである。

【図 1 1】

図 1 1 は、攪乱用データおよび処理済攪乱用データ生成方法の例を示すフローチャートである。

【図 1 2】

図 1 2 は、攪乱用データおよび処理済攪乱用データ生成方法の例を示すフローチャートである。

【図 1 3】

図 1 3 は、表引きによるデータ処理を、2つの攪乱用データを用いて変形して処理する方法の例を示すフローチャートである。

【図 1 4】

図 1 4 は、攪乱用データおよび変形済み表生成方法の例を示すフローチャートである。

【図 1 5】

図 1 5 は、攪乱用データおよび変形済み表生成方法の例を示すフローチャートである。

【図 1 6】

図 1 6 は、攪乱用データおよび変形済み表生成方法。

【図 1 7】

表引きと1つのデータ処理手段からなるデータ処理を、2つの攪乱用データを用いて変形して処理する方法の例を示すフローチャートである。

【図 1 8】

図 1 8 は、攪乱用データおよび変形済み表生成方法の例を示すフローチャートである。

【図 1 9】

図 1 9 は、攪乱用データおよび変形済み表生成方法の例を示すフローチャートである。

【図 2 0】

図 2 0 は、攪乱用データおよび変形済み表生成方法の例を示すフローチャートである。

【図 2 1】

図 2 1 は、攪乱用データおよび変形済み表生成方法の例を示すフローチャートである。

【図 2 2】

図 2 2 は、攪乱用データおよび変形済み表生成方法の例を示すフローチャートである。

【図 2 3】

図 2 3 は、表引きと 1 つのデータ処理手段からなる処理を、2 回繰り返すデータ処理を、2 つの攪乱用データを用いて変形して処理する方法の例を示すフローチャートである。

【図 2 4】

図 2 4 は、表引きと 1 つのデータ処理手段からなる処理を、2 回繰り返すデータ処理を、4 つの攪乱用データを用いて変形して処理する方法の例を示すフローチャートである。

【図 2 5】

図 2 5 は、攪乱用データおよび変形済み表生成方法の例を示すフローチャートである。

【図 2 6】

図 2 6 は、攪乱用データおよび変形済み表生成方法の例を示すフローチャートである。

【図 2 7】

図 2 7 は、攪乱用データおよび変形済み表生成方法の例を示すフローチャートである。

【図 2 8】

図 2 8 は、攪乱用データおよび変形済み表生成方法の例を示すフローチャートである。

【図 2 9】

図 2 9 は、DES 処理用 SBOX 攪乱用データおよび変形 SBOX 生成方法の例を示すフローチャートである。

【図 3 0】

図 3 0 は、D E S 処理用平文攪乱用データ生成方法の例を示すフローチャートである。

【図 3 1】

図 3 1 は、D E S 処理用秘密鍵攪乱用データ生成方法の例を示すフローチャートである。

【図 3 2】

図 3 2 は、平文変形方法の例を示すフローチャートである。

【図 3 3】

図 3 3 は、秘密鍵変形方法の例を示すフローチャートである。

【図 3 4】

図 3 4 は、D E S 第 1、第 5、第 9、第 1 3 ラウンド処理方法の例を示すフローチャートである。

【図 3 5】

図 3 5 は、D E S 第 2、第 6、第 1 0、第 1 4 ラウンド処理方法の例を示すフローチャートである。

【図 3 6】

図 3 6 は、D E S 第 3、第 7、第 1 1、第 1 5 ラウンド処理方法の例を示すフローチャートである。

【図 3 7】

図 3 7 は、D E S 第 4、第 8、第 1 2 ラウンド処理方法の例を示すフローチャートである。

【図 3 8】

図 3 8 は、D E S 第 1 6 ラウンド処理方法の例を示すフローチャートである。

【図 3 9】

図 3 9 は、D E S 最終逆変形方法の例を示すフローチャートである。

【図 4 0】

図 4 0 は、D E S 第 1、第 5、第 9、第 1 3 ラウンド処理方法の例を示すフローチャートである。

【図 4 1】

図 4 1 は、DES 第 2、第 6、第 1 0、第 1 4 ラウンド処理方法の例を示すフローチャートである。

【図 4 2】

図 4 2 は、DES 第 3、第 7、第 1 1、第 1 5 ラウンド処理方法の例を示すフローチャートである。

【図 4 3】

図 4 3 は、DES 第 4、第 8、第 1 2 ラウンド処理方法の例を示すフローチャートである。

【図 4 4】

図 4 4 は、DES 第 1 6 ラウンド処理方法の例を示すフローチャートである。

【図 4 5】

図 4 5 は、SBOX アクセス方法の例を示すフローチャートである。

【図 4 6】

図 4 6 は、SBOX 表の例を示す図である。

【図 4 7】

図 4 6 は、SBOX 表変形方法の例を示すフローチャートである。

【図 4 8】

図 4 8 は、ハミングウェイト一定攪乱用かつ処理済のハミングウェイトも一定のデータ生成方法の例を示すフローチャートである。

【図 4 9】

図 4 9 は、ハミングウェイト一定乱数生成方式の例を示すフローチャートである。

【図 5 0】

図 5 0 は、ハミングウェイト一定乱数生成方式の例を示すフローチャートである。

【図 5 1】

図 5 1 は、ハミングウェイト一定乱数生成方式の例を示すフローチャートである。

【図 5 2】

図 5 2 は、D E S 処理用 S B O X 攪乱用データおよび変形 S B O X 生成方法の例を示すフローチャートである。

【図 5 3】

図 5 3 は、D E S 処理用攪乱用データ生成方法の例を示すフローチャートである。

【図 5 4】

図 5 4 は、D E S 処理中間データ攪乱方法の例を示すフローチャートである

【図 5 5】

図 5 5 は、図 1 2 に対応するテーブルの例を示す図である。

【図 5 6】

図 5 6 は、第 1 攪乱用データ格納手段（1 5 0 1）に格納された第 1 攪乱用データ、及び第 2 攪乱用データ格納手段（1 5 0 2）に格納された第 2 攪乱用データの例を示す図である。

【図 5 7】

図 5 7 は、表格納手段（1 5 0 7）に格納されたデータの例を示す図である。

【図 5 8】

図 5 8 は、第 1 攪乱用データ、第 2 攪乱用データ及び変形済み表のテーブルの例を示す図である。

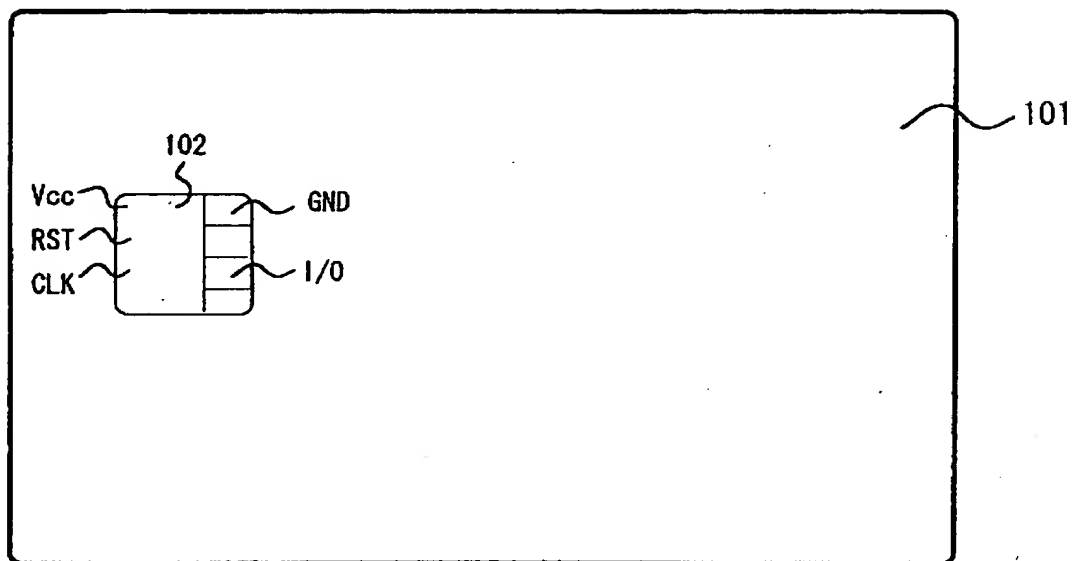
【符号の説明】

4 0 1 : 入力データ、4 0 2 : データ変形処理手段、4 0 3 : 攪乱用データ、4 0 4 : 変形データ、4 0 5 : 変形データ処理手段、4 0 6 : 変形データ、4 0 7 : 変形データ、4 0 8 : 変形済み攪乱用データ、4 0 9 : 処理済データ、5 0 2 : ハミングウェイト一定の攪乱用データ、5 0 3 : 攪乱用データ処理手段、5 0 4 : 処理済攪乱用データ、5 0 5 : ハミングウェイト検査手段、6 0 1 : n ビット乱数発生手段、6 0 2 : n ビット乱数、6 0 3 : ビット反転処理手段、6 0 4 : 反転 n ビット乱数、6 0 5 : データ統合手段、6 0 6 : ハミングウェイト一定 2 n ビット乱数。

【書類名】 図面

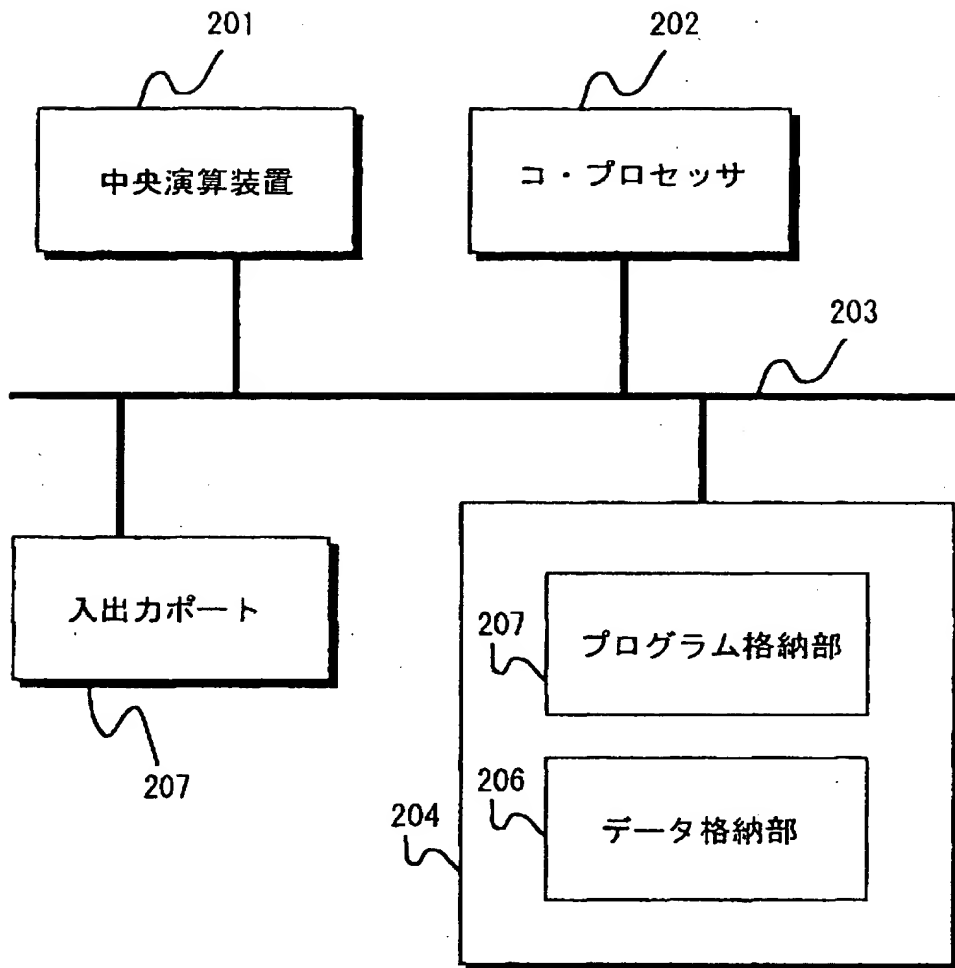
【図 1】

図 1



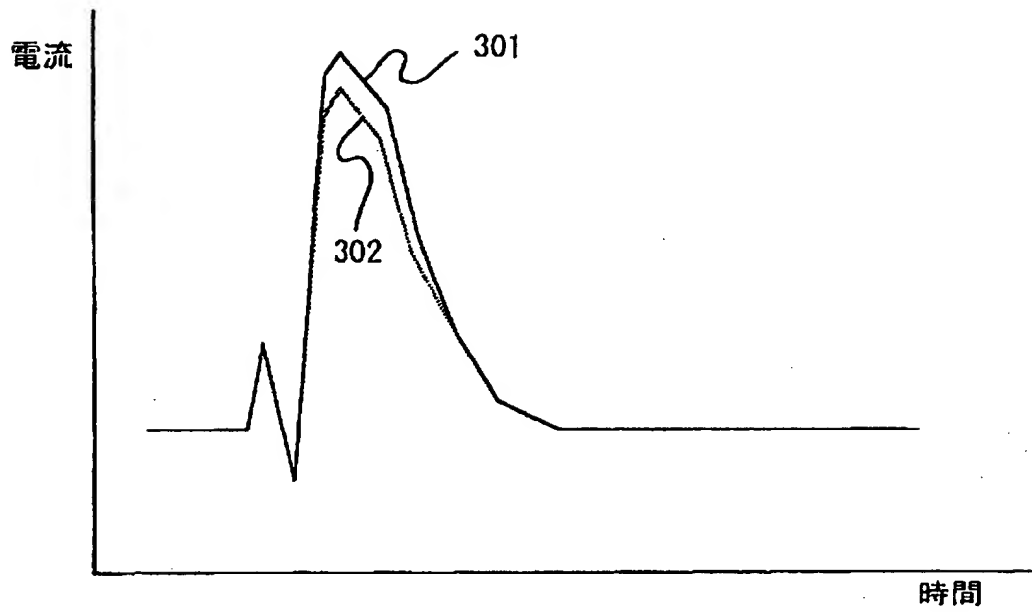
【図 2】

図 2



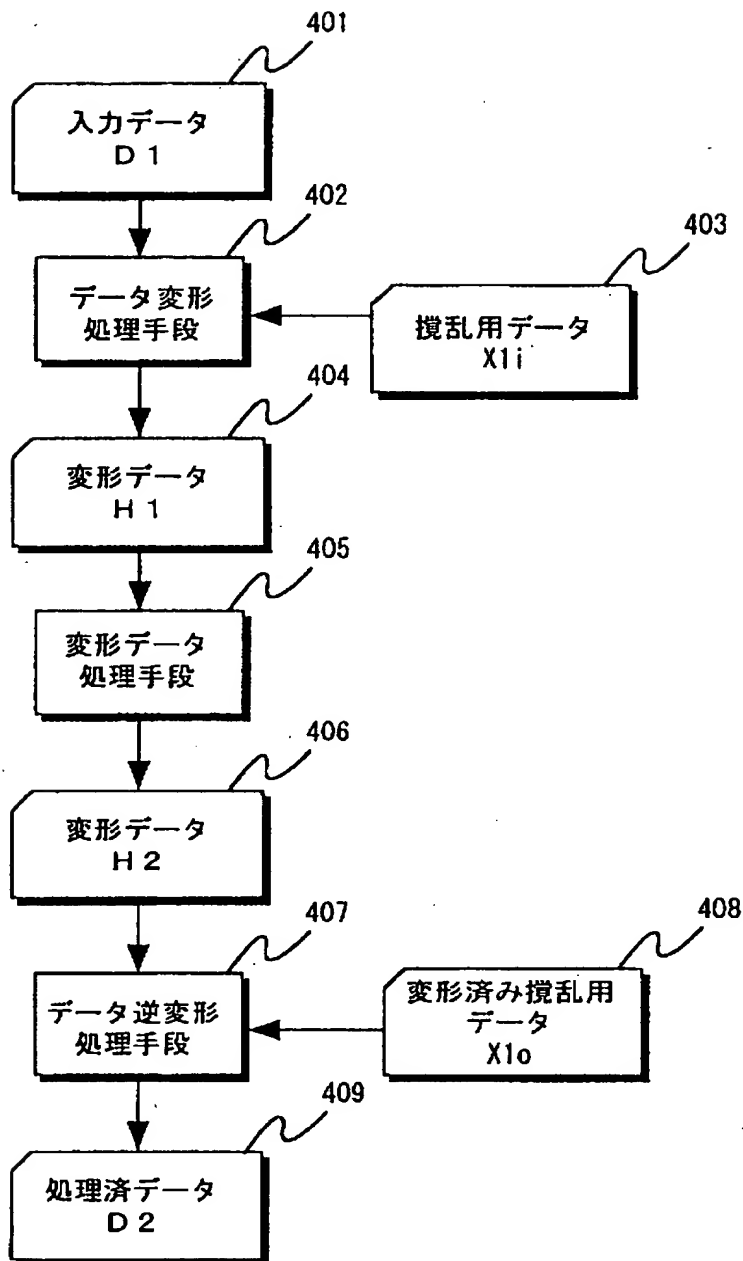
【図 3】

図 3



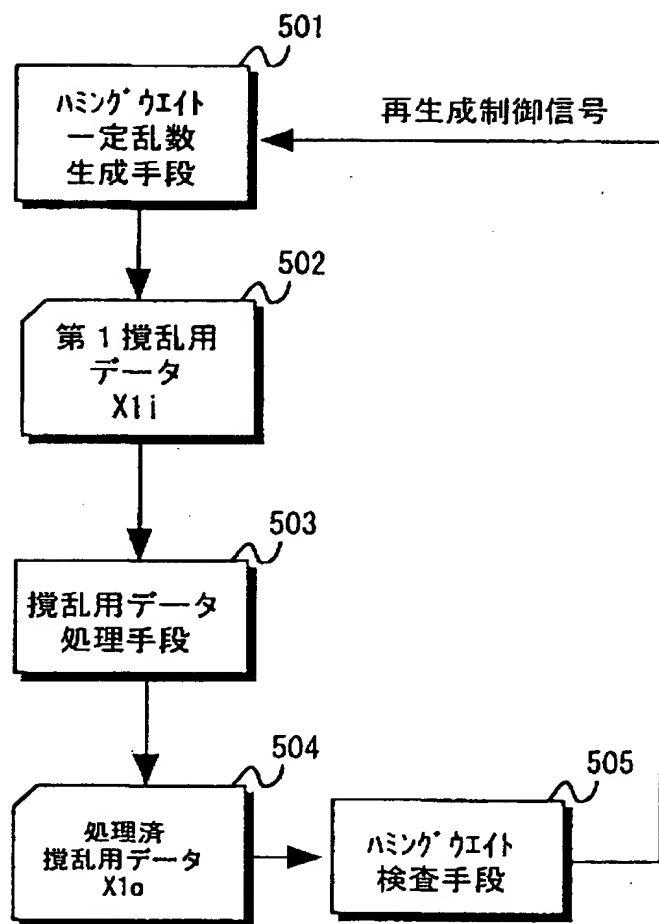
【図 4】

図 4



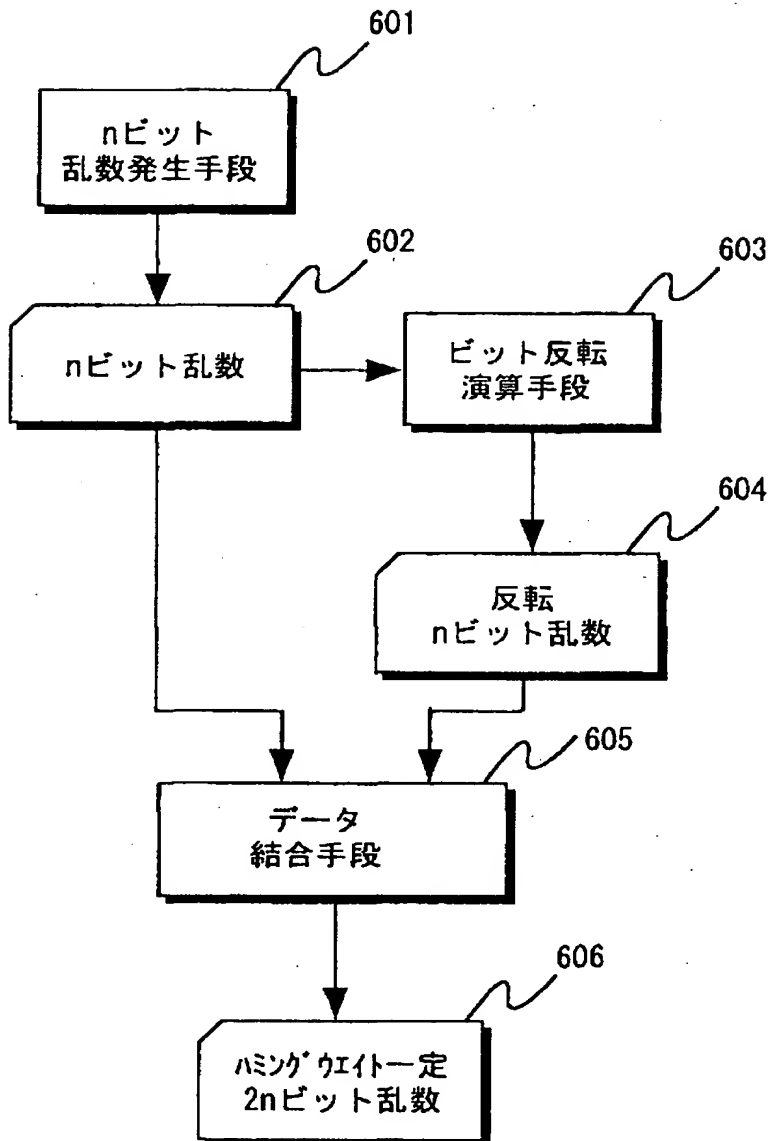
【図 5】

図 5

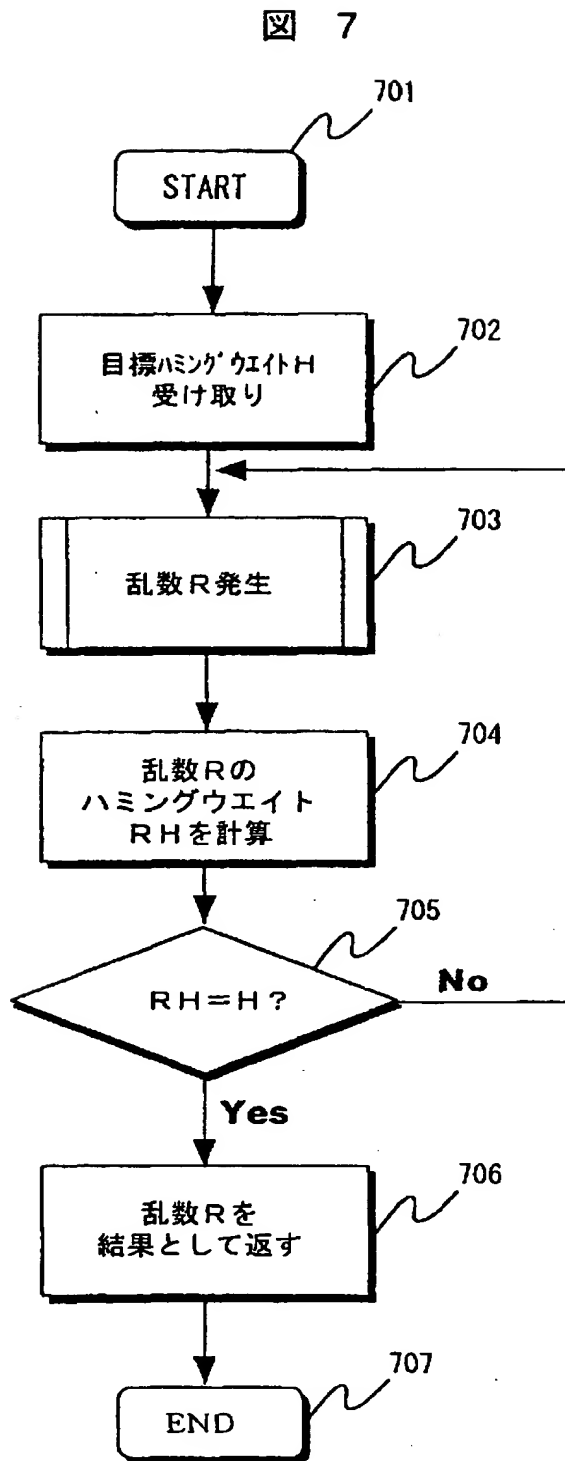


【図 6】

図 6

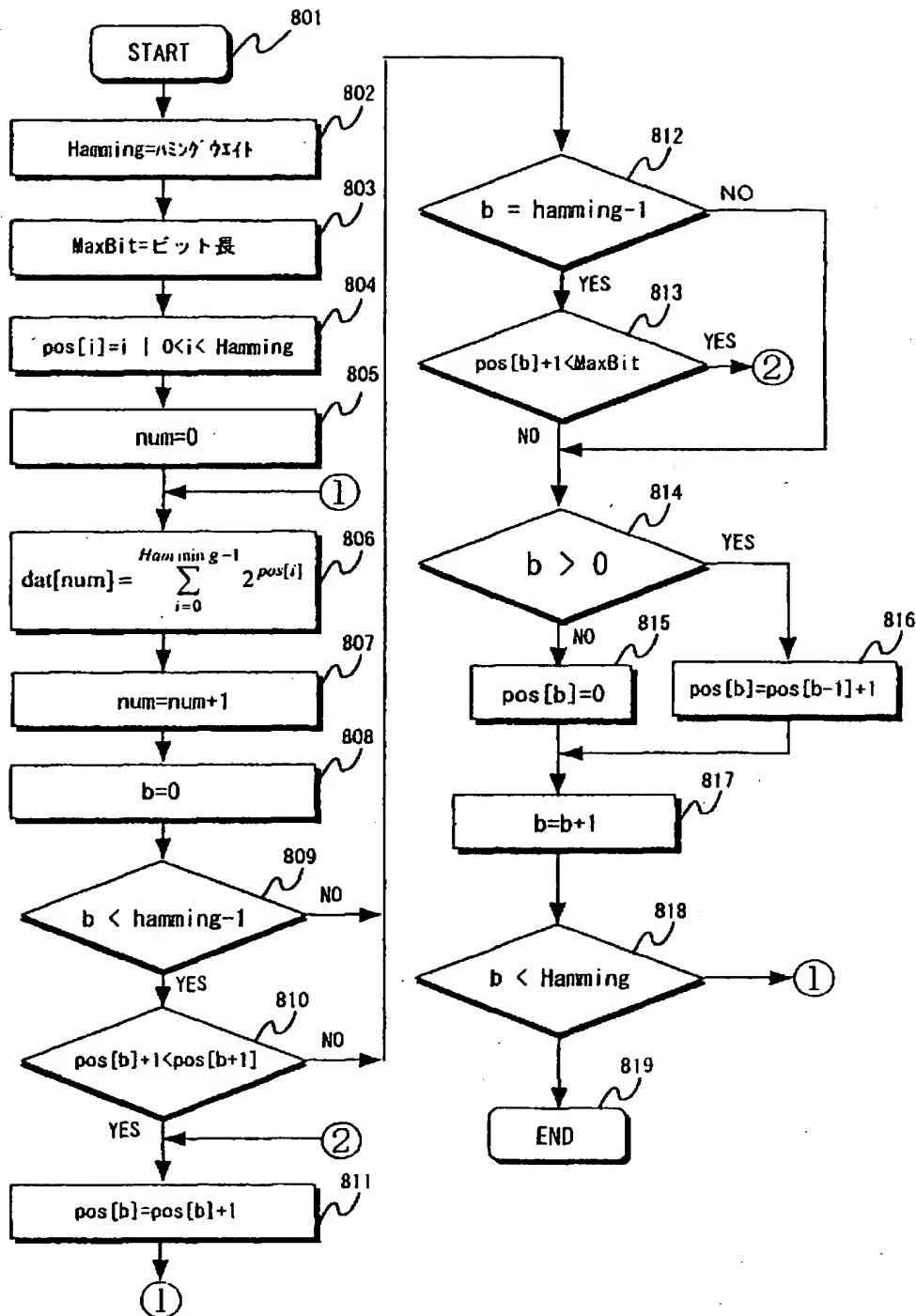


【図 7】



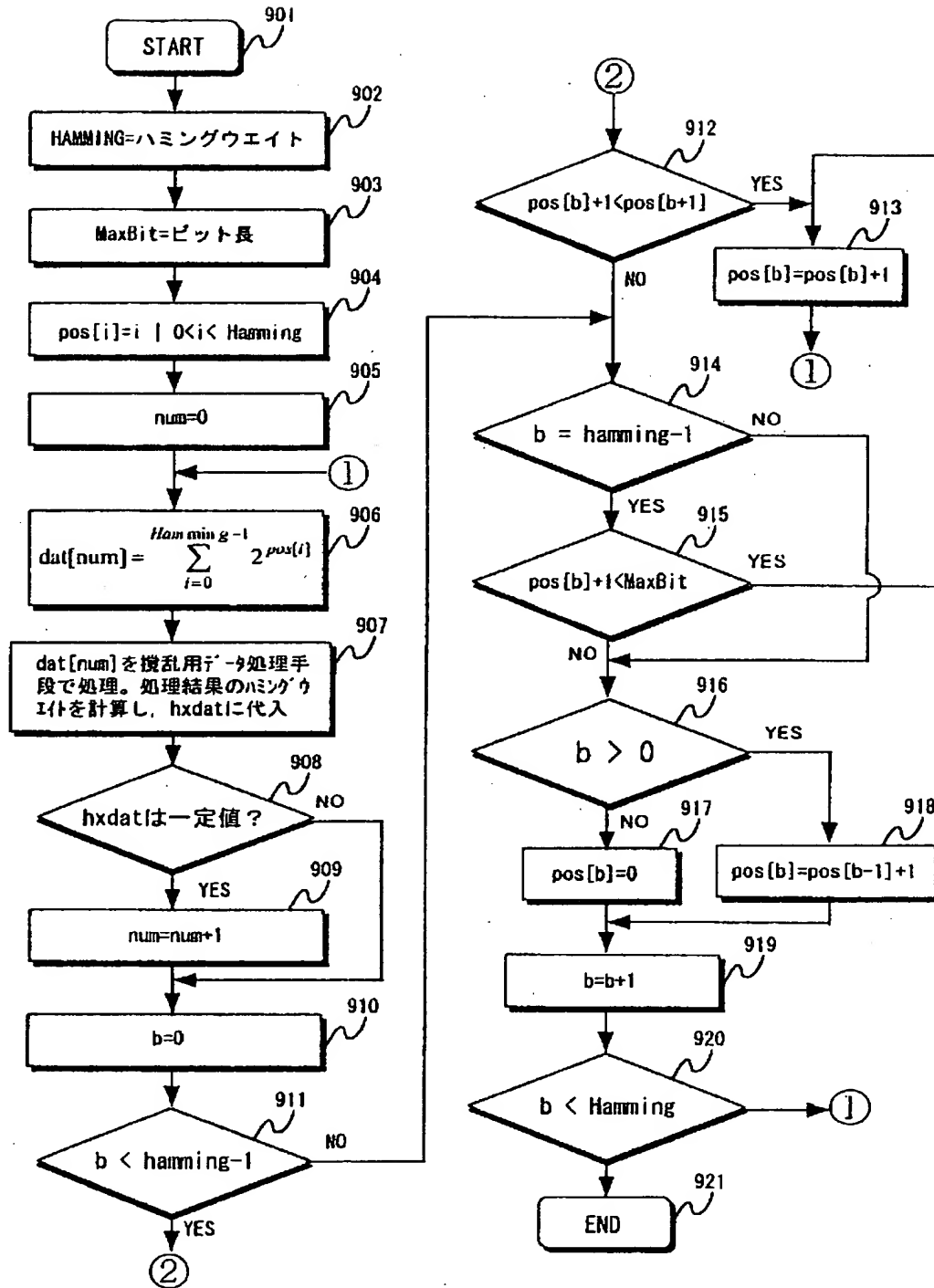
【図 8】

図 8

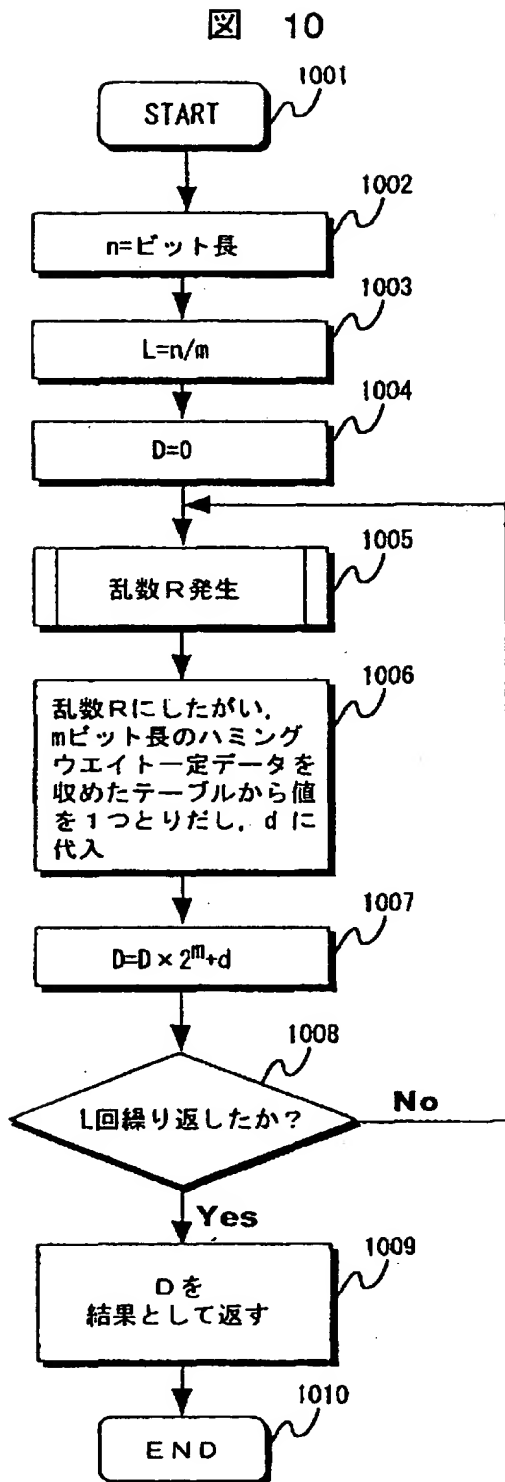


【図 9】

図 9

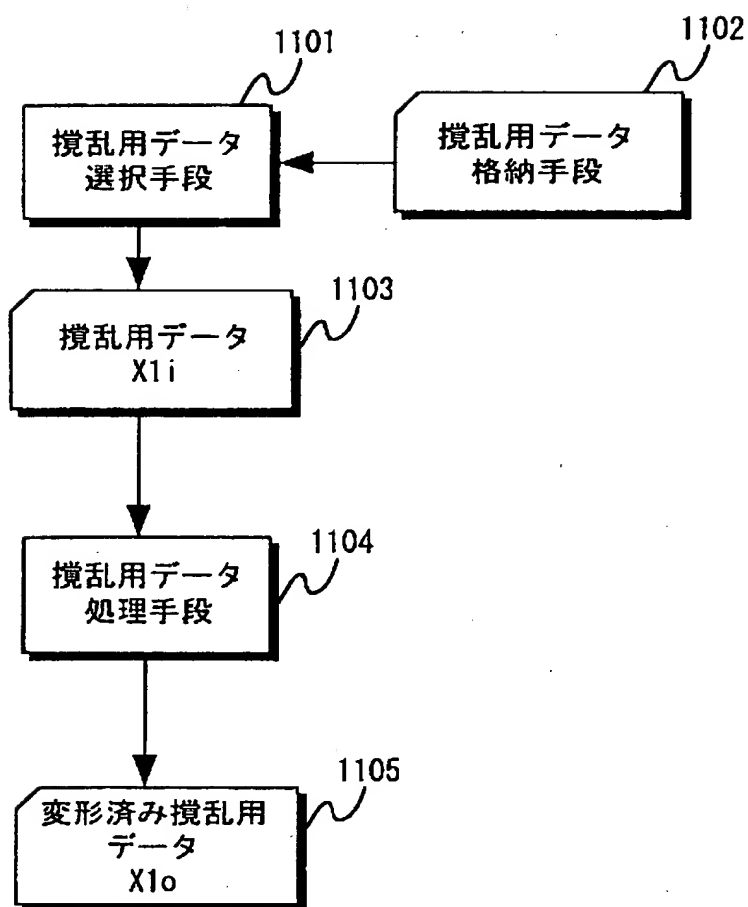


【図 10】



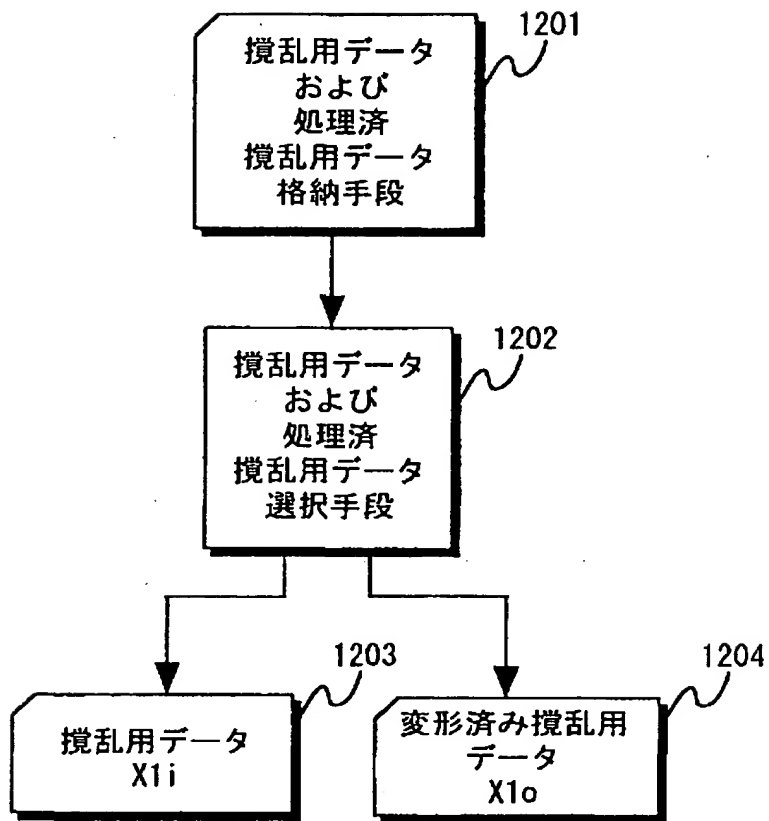
【図 11】

図 11



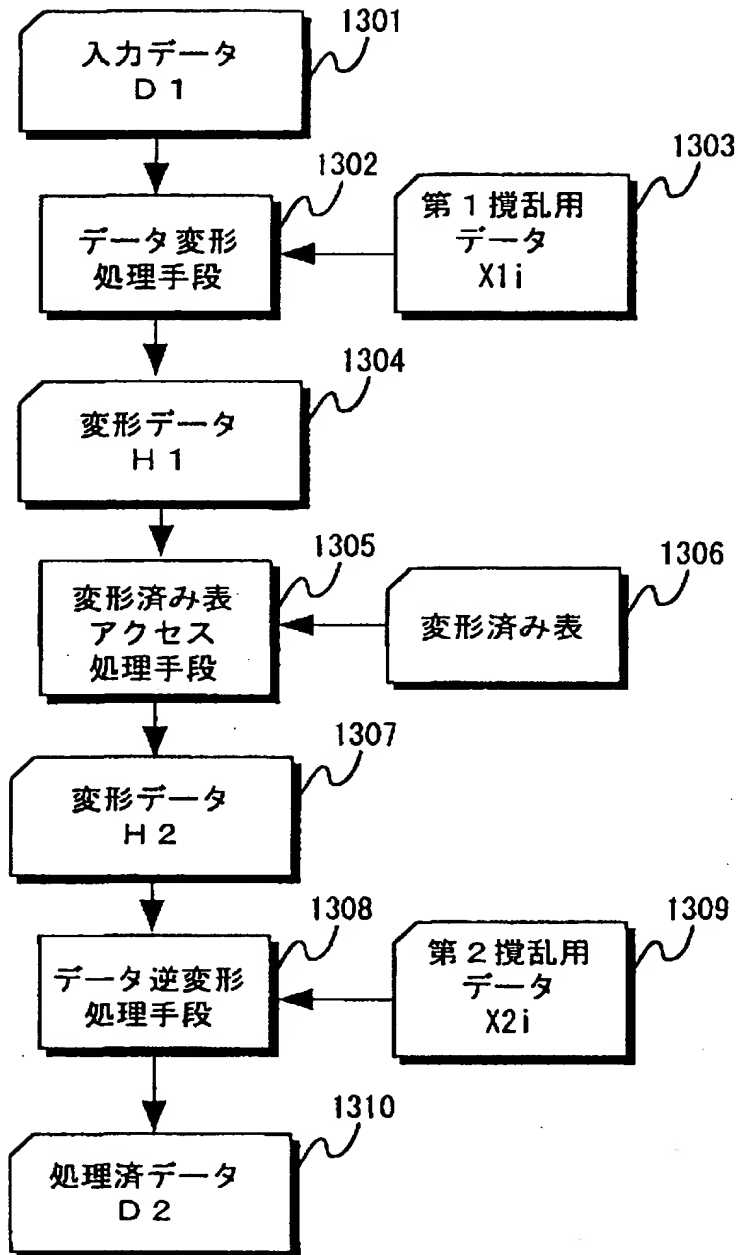
【図 12】

図 12



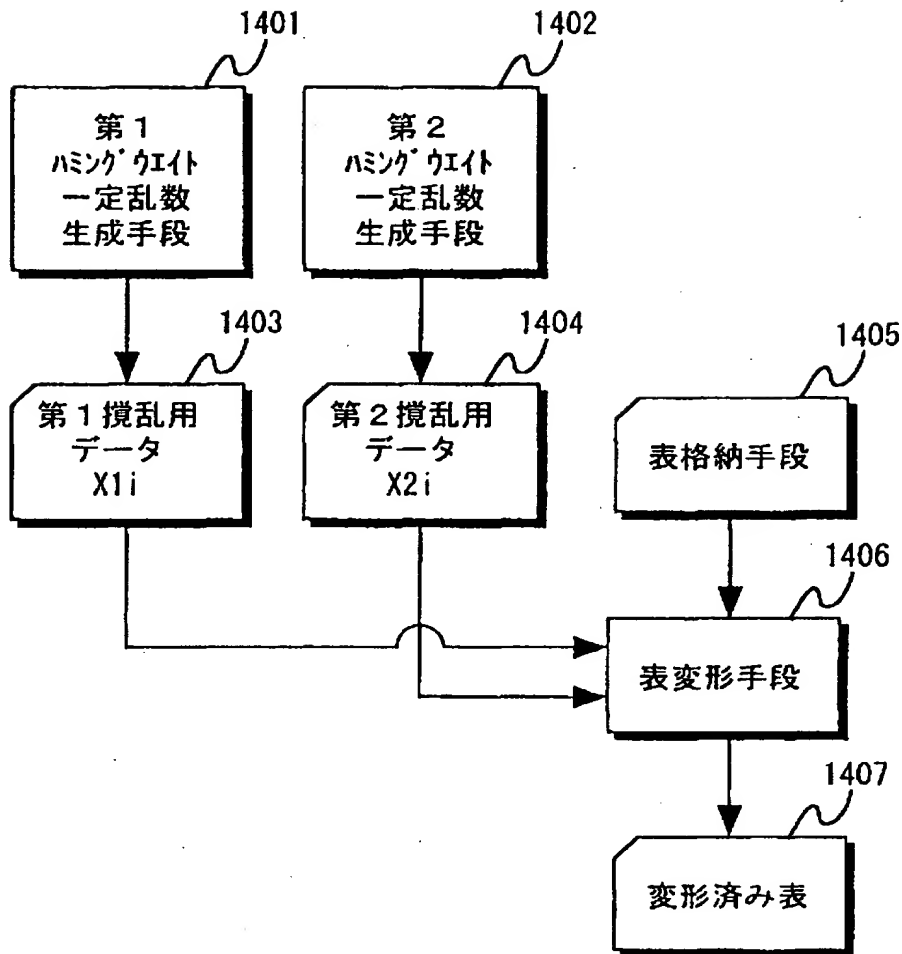
【図 13】

図 13



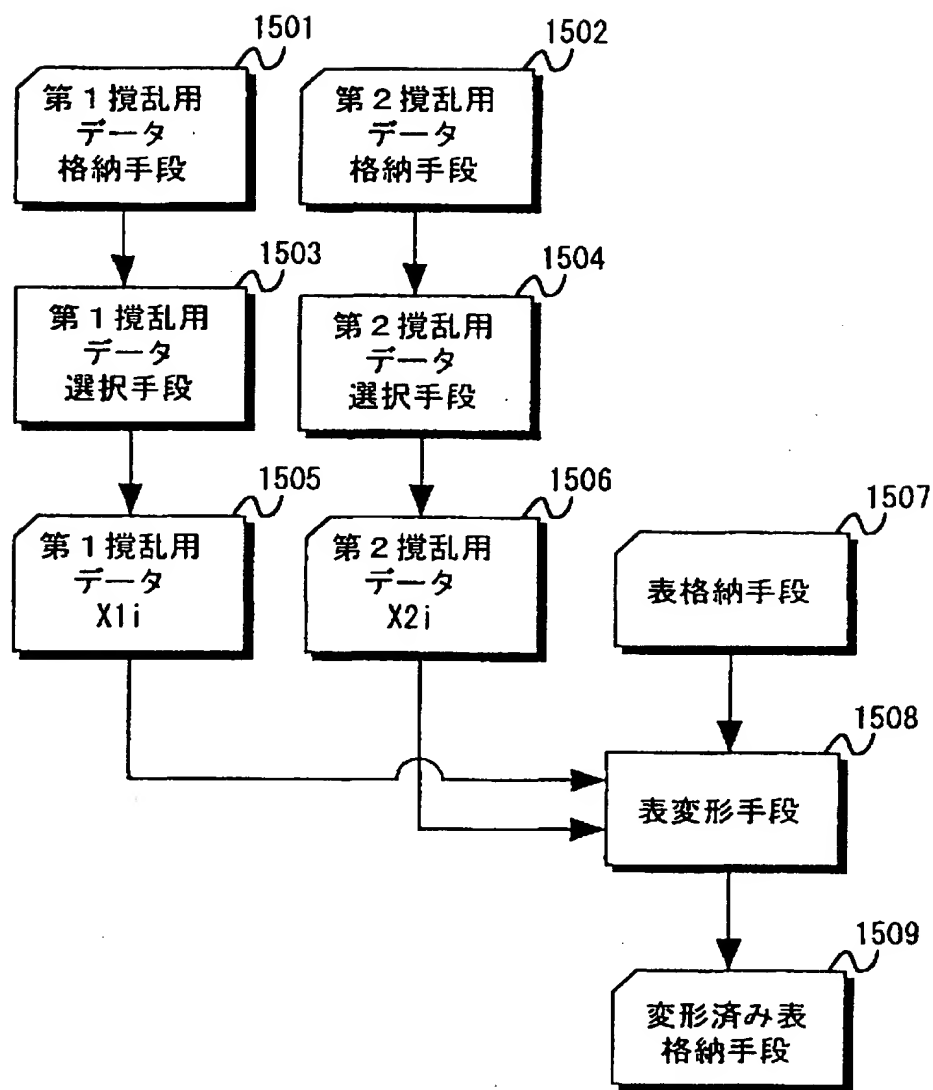
【図 14】

図 14



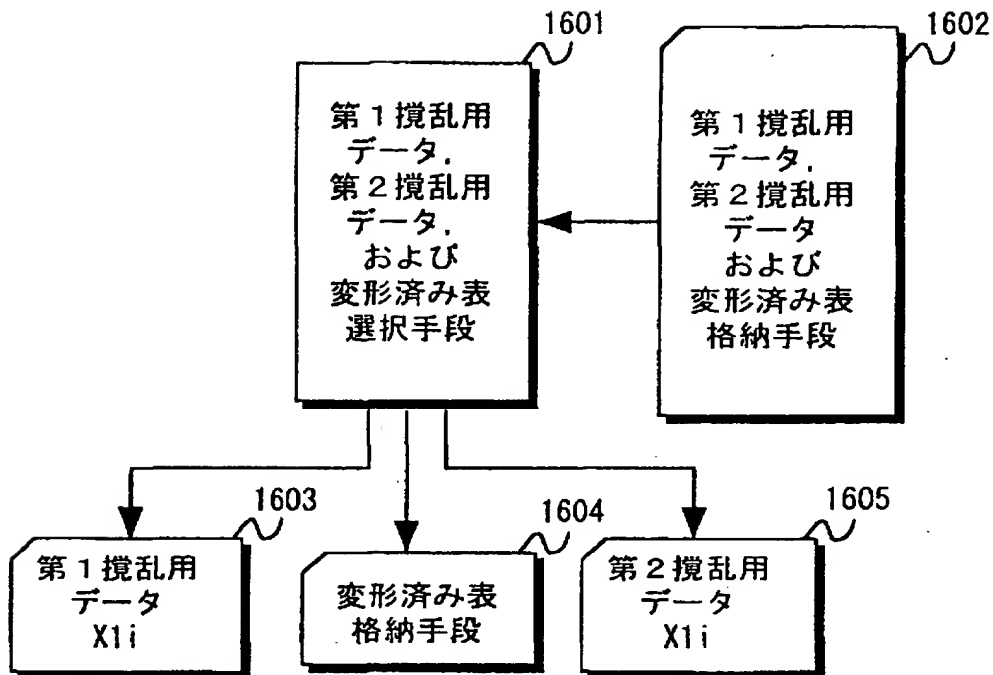
【図 15】

図 15



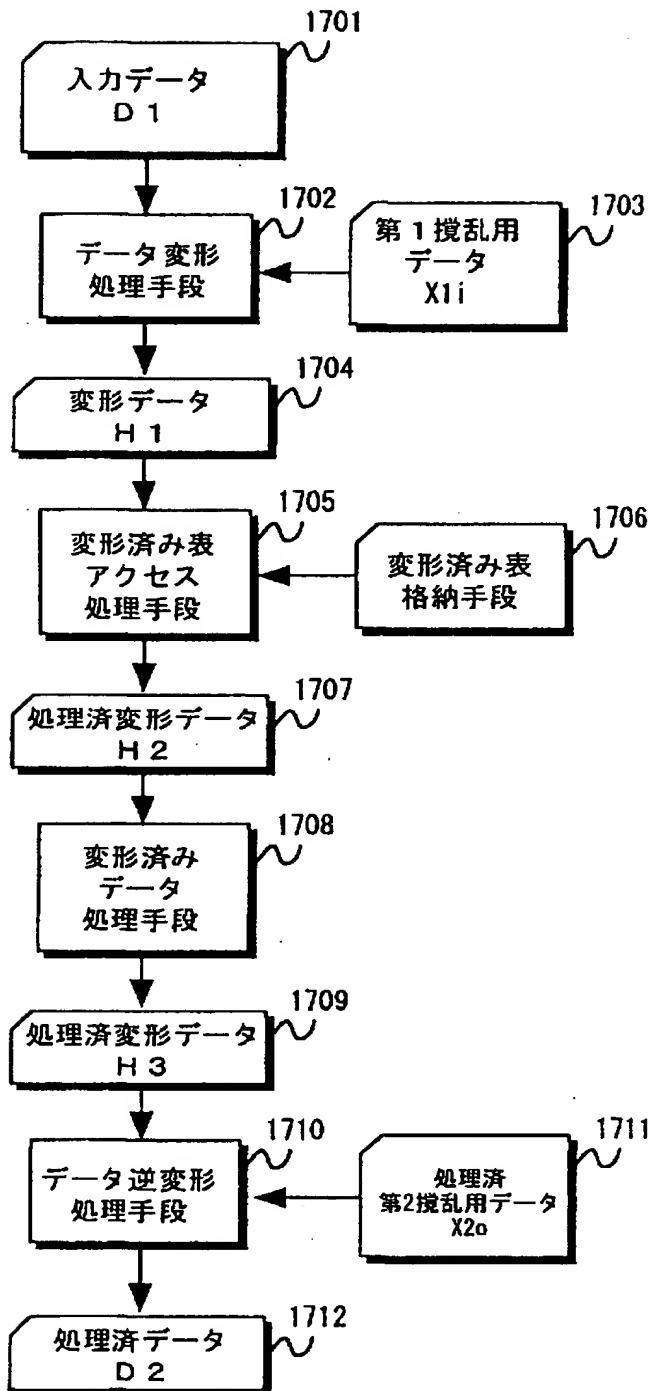
【図 16】

図 16



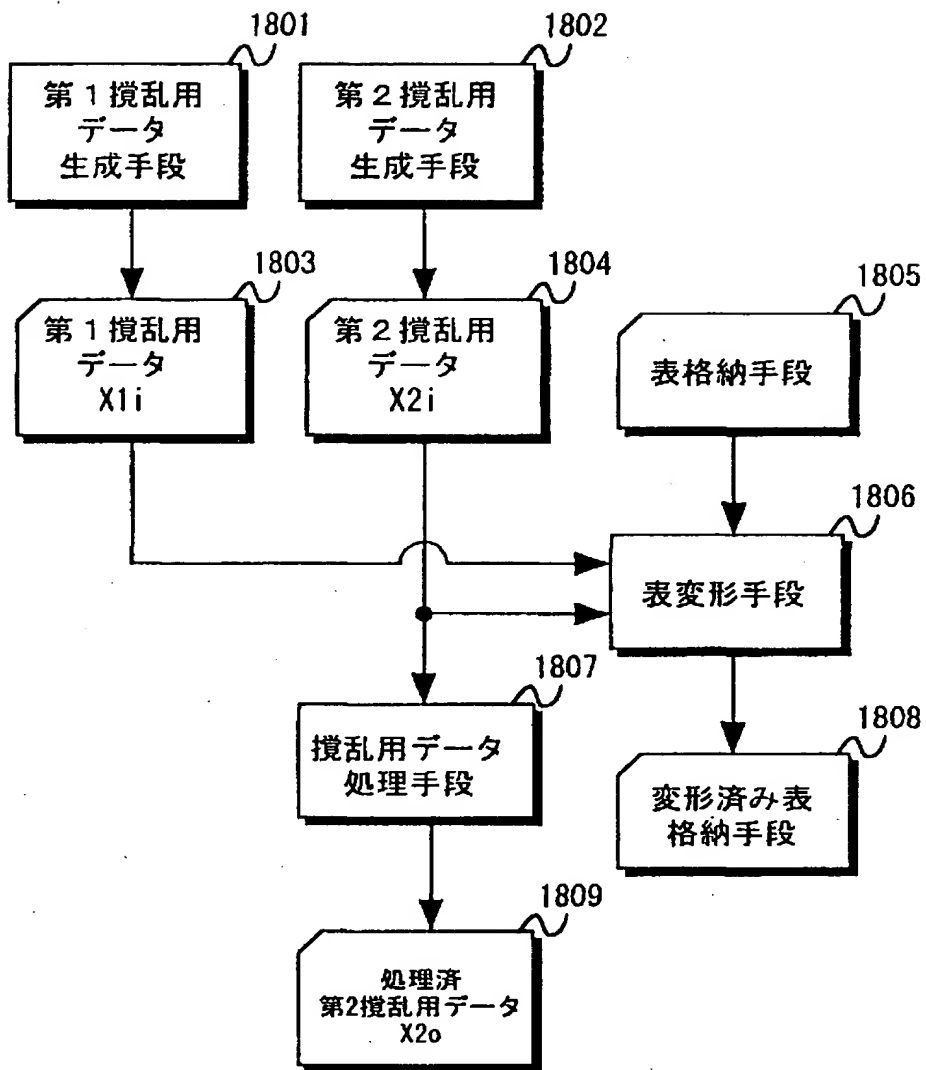
【図 17】

図 17



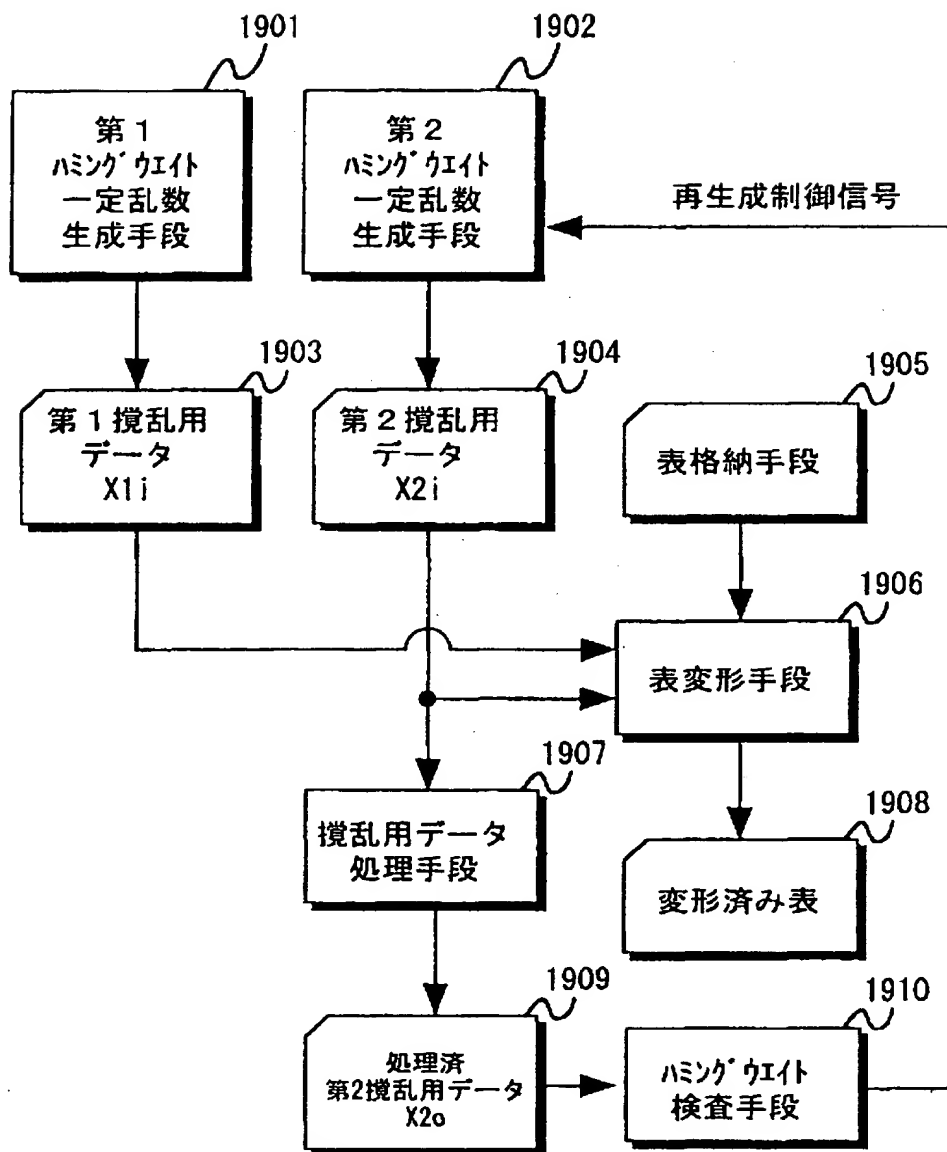
【図 18】

図 18



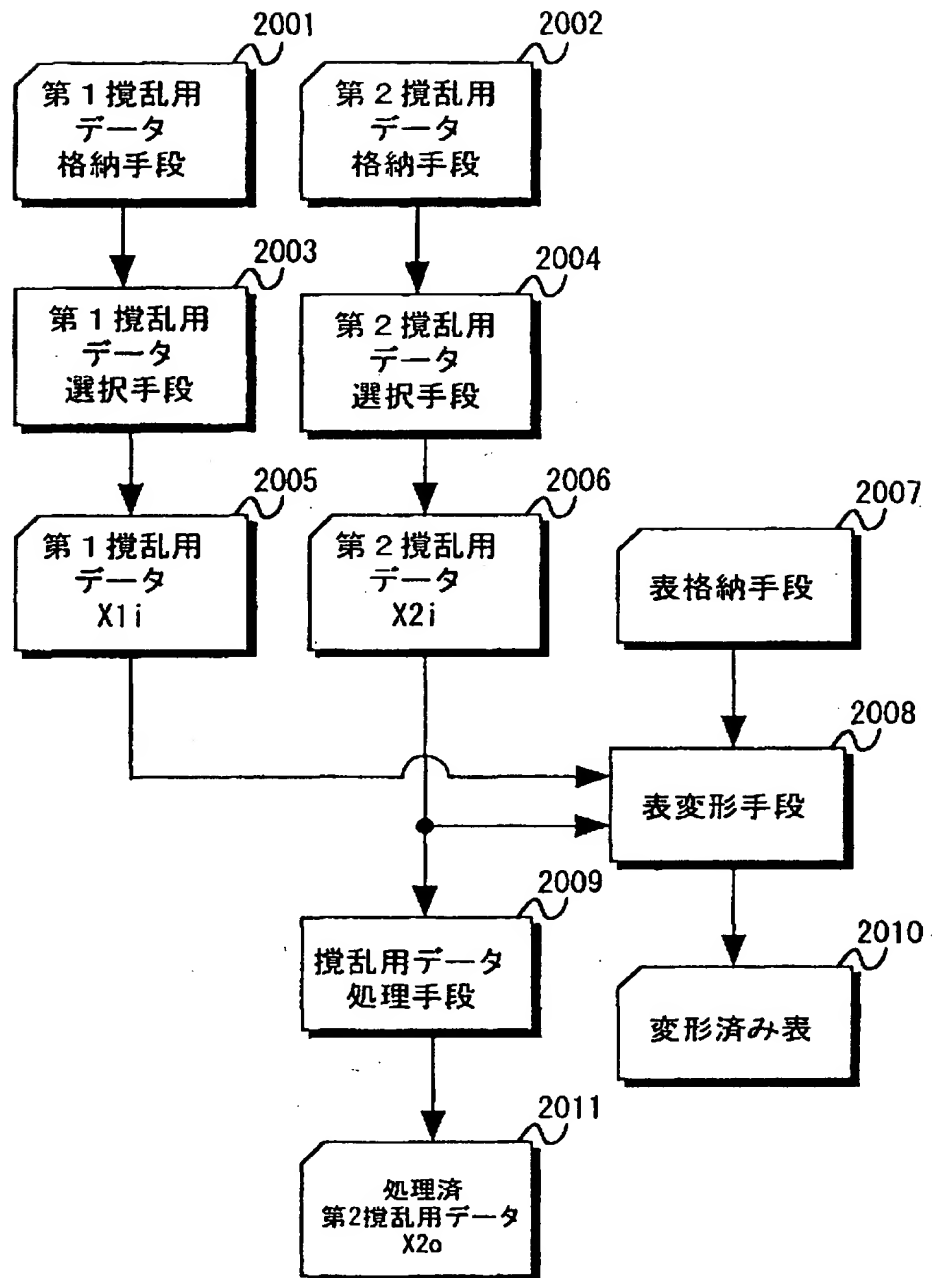
【図 1 9】

図 19



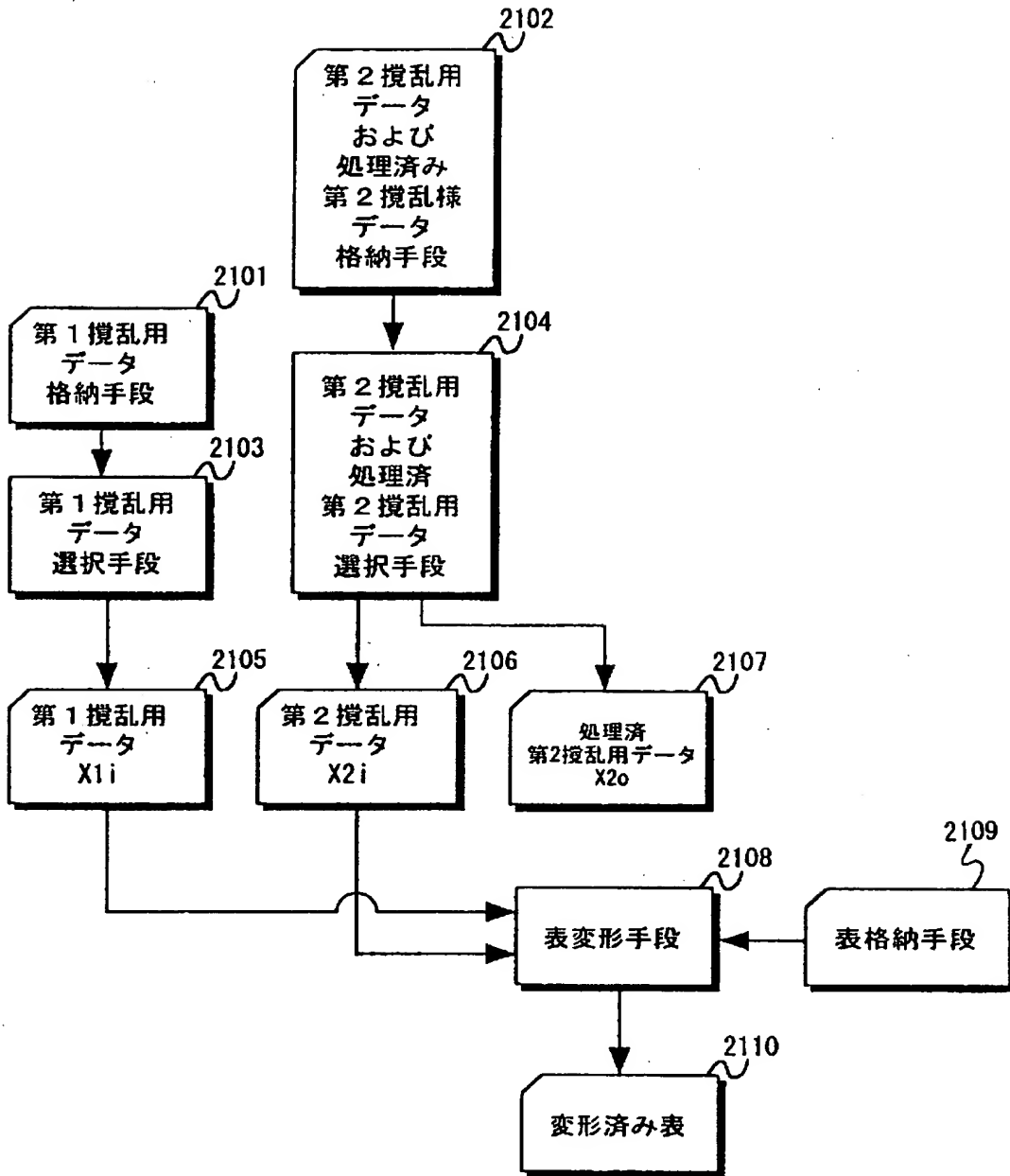
【図 2 0】

図 20



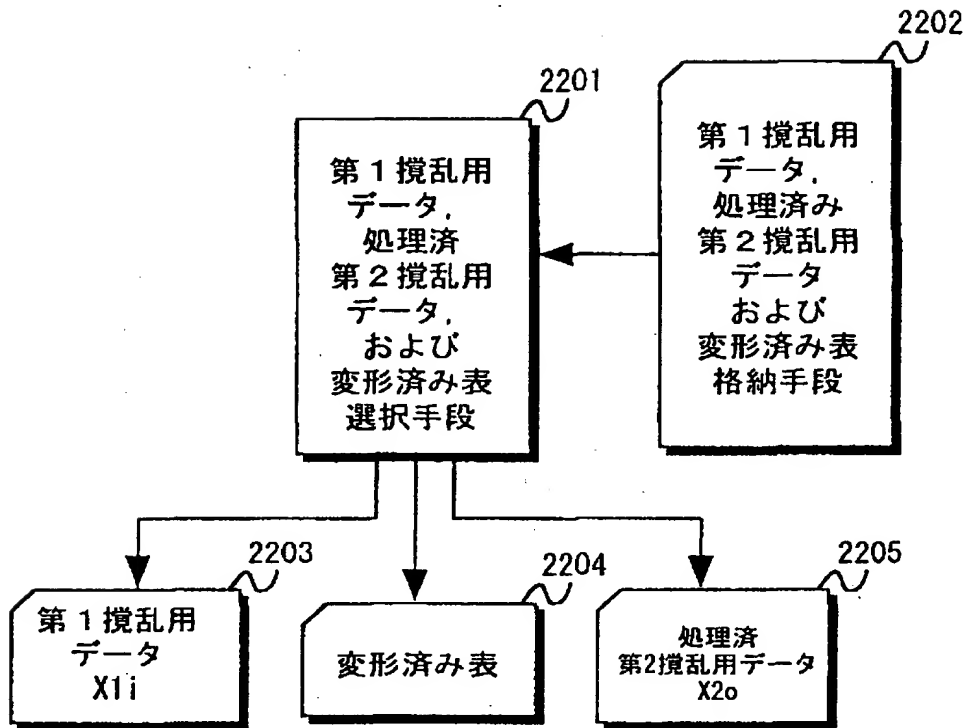
【図 21】

図 21



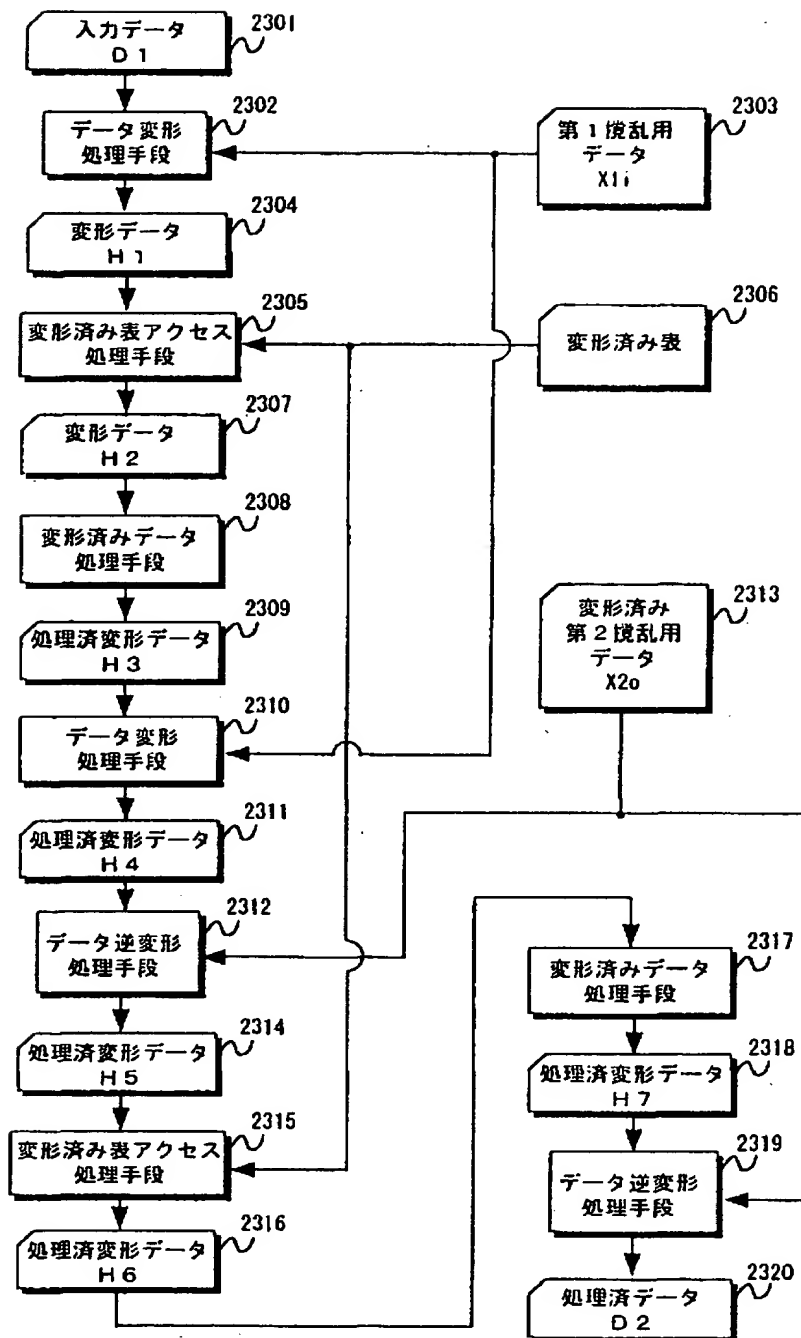
【図 22】

図 22



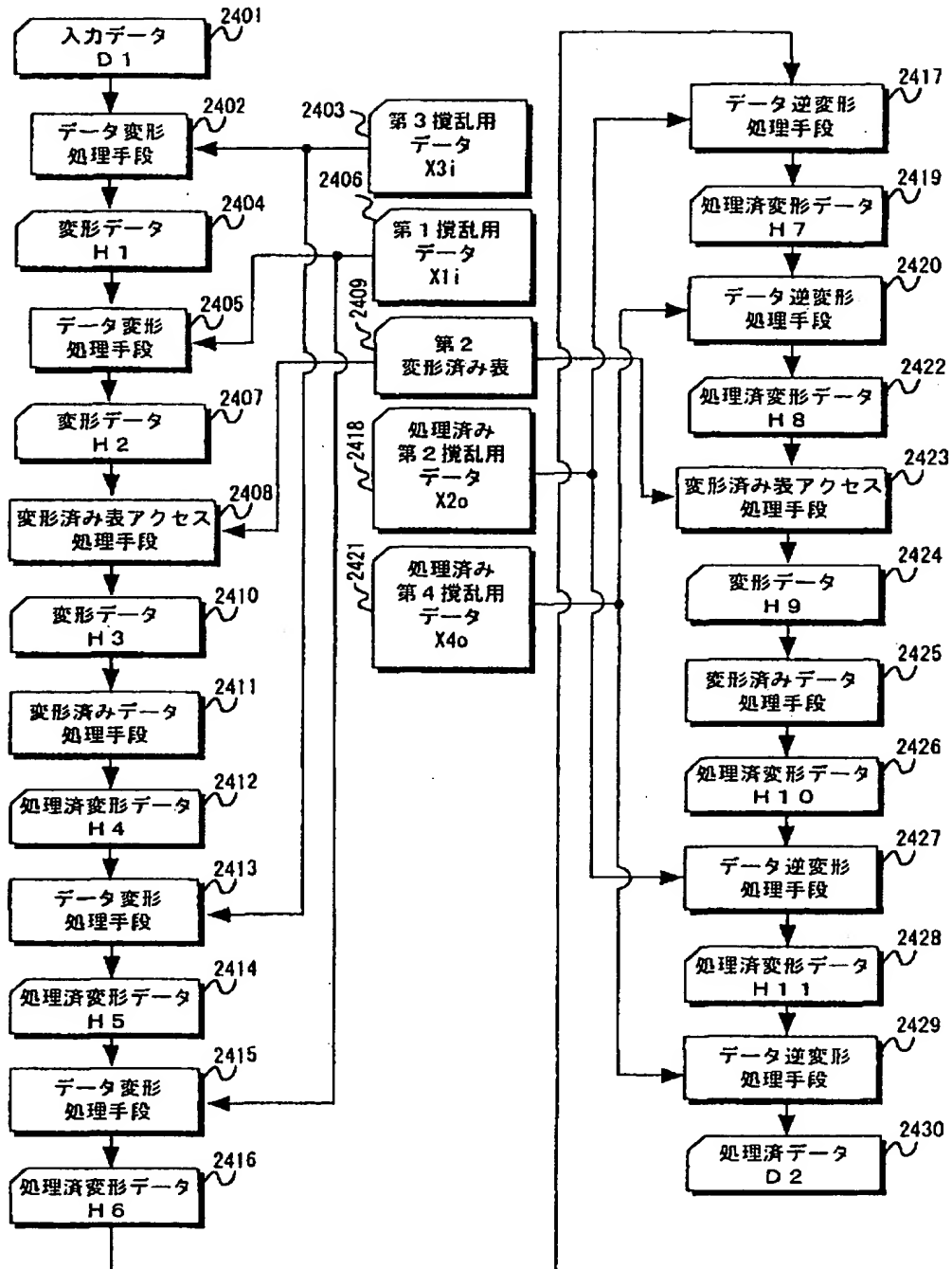
【図 23】

図 23



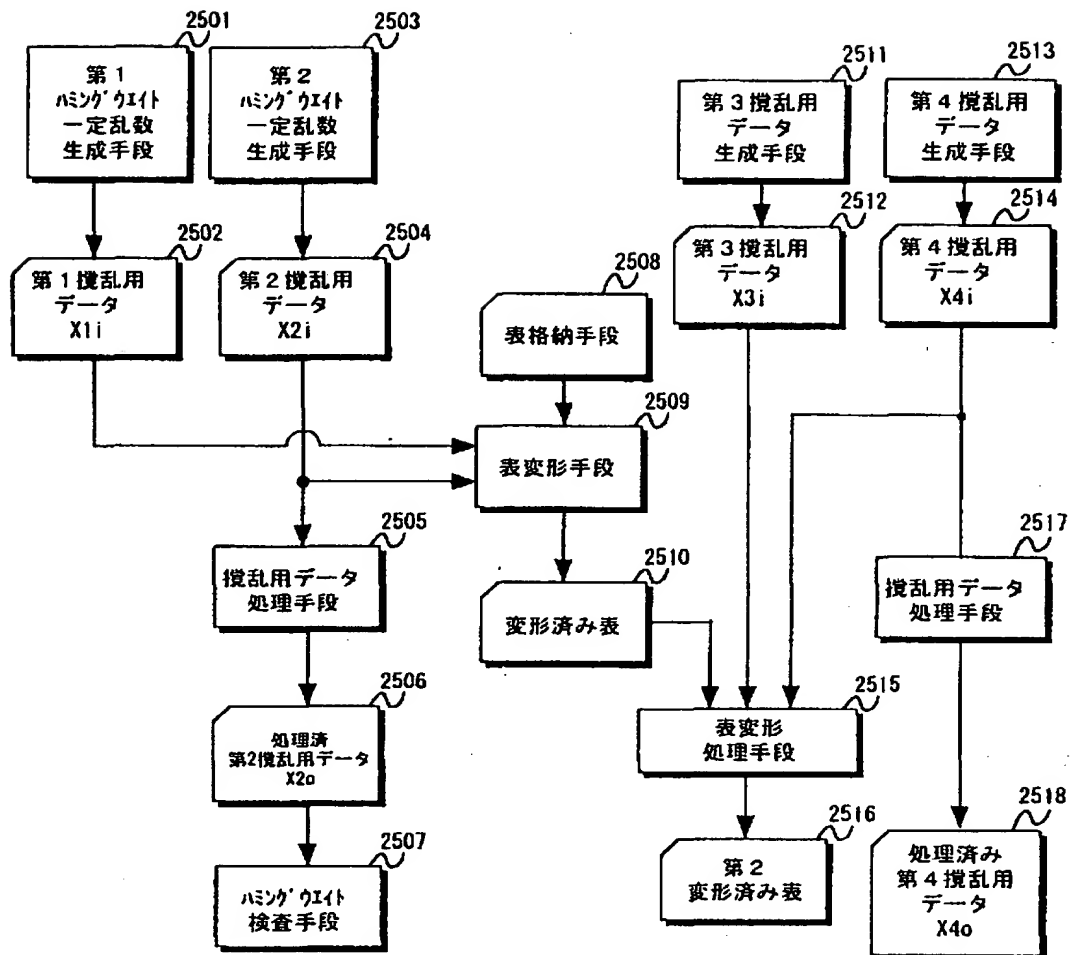
【図 24】

図 24



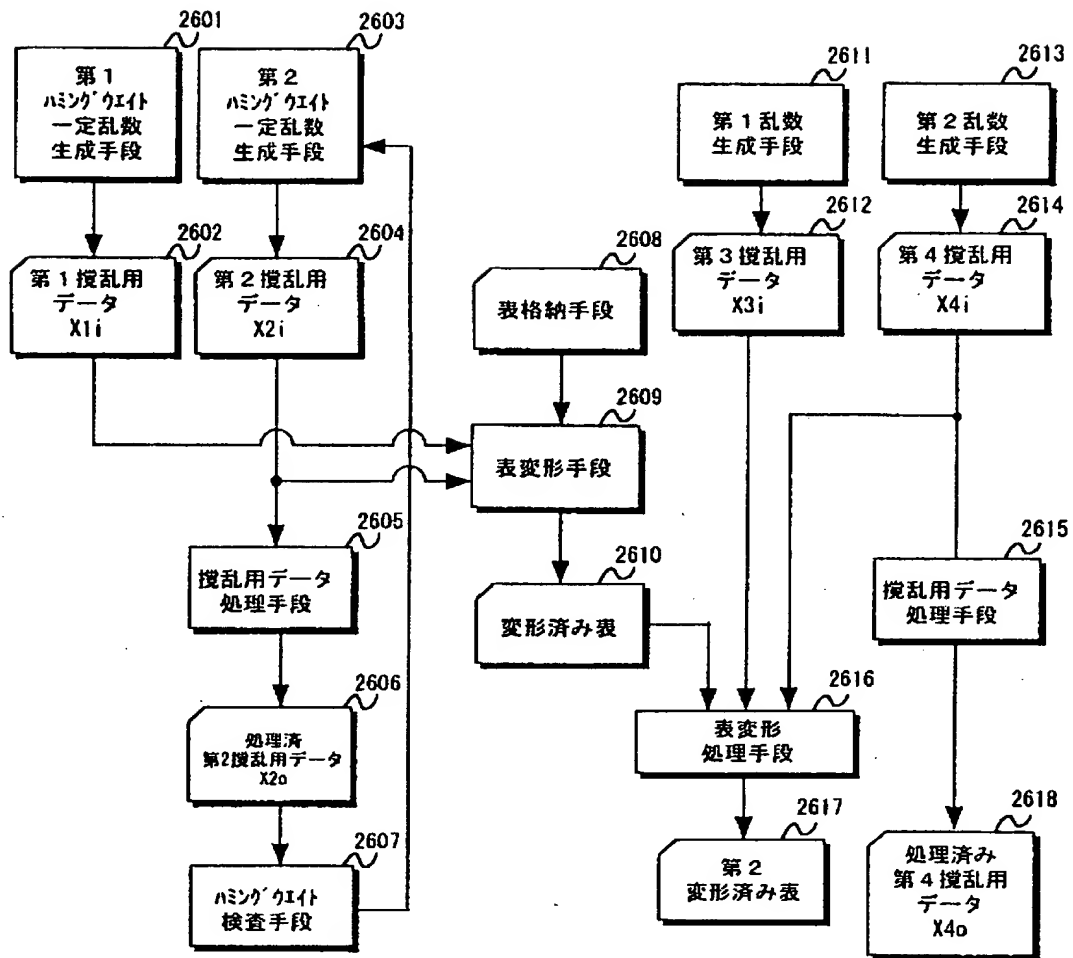
【図 25】

図 25



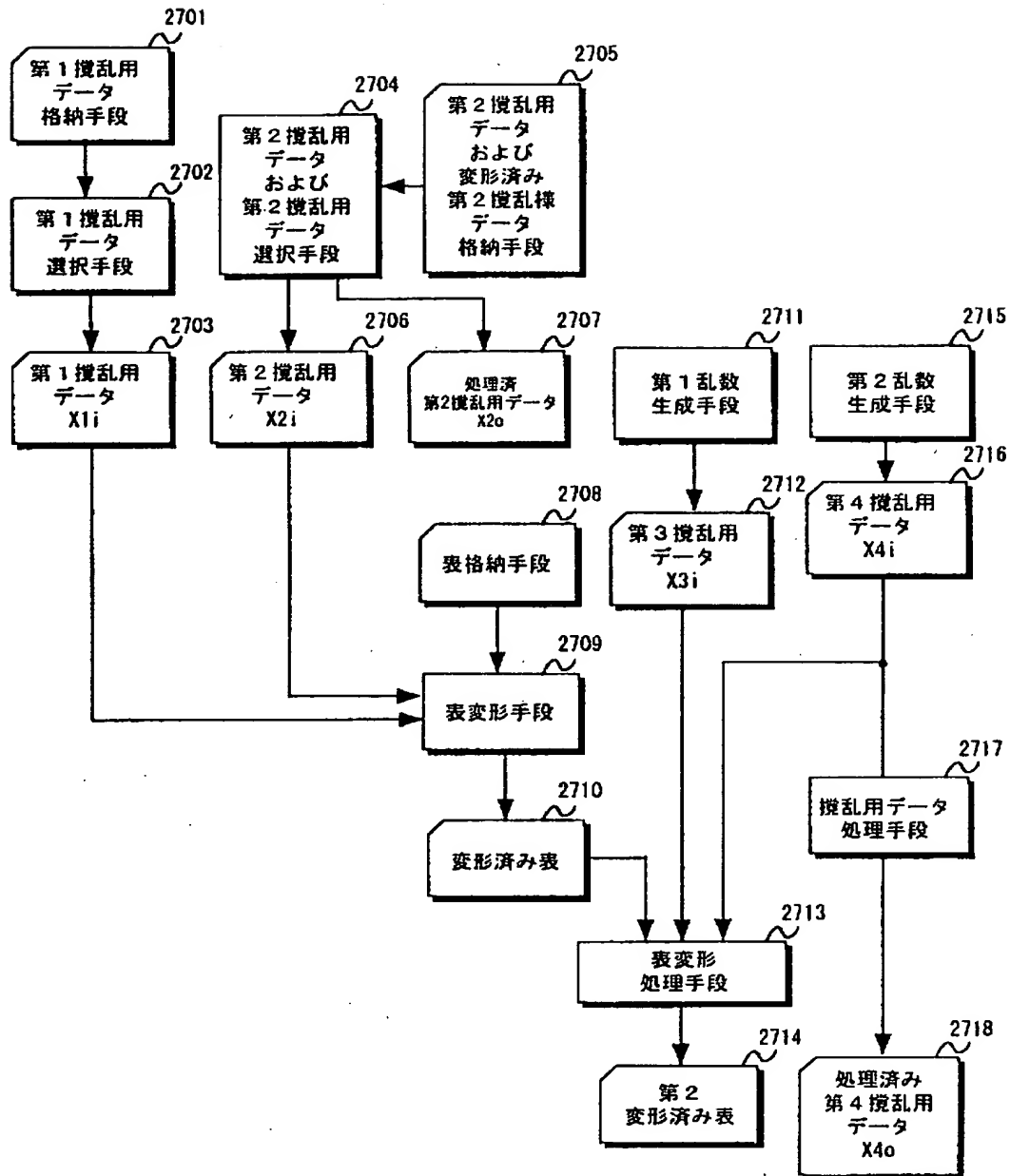
【図 26】

図 26



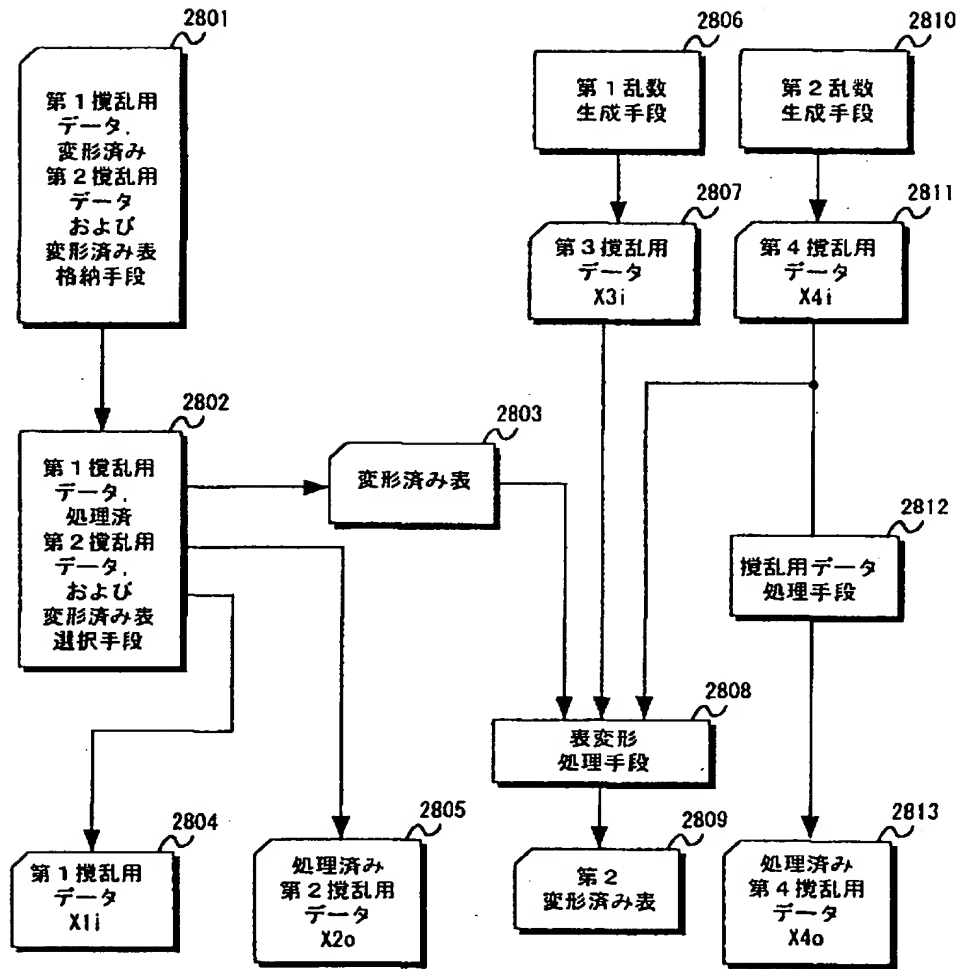
【図 27】

図 27



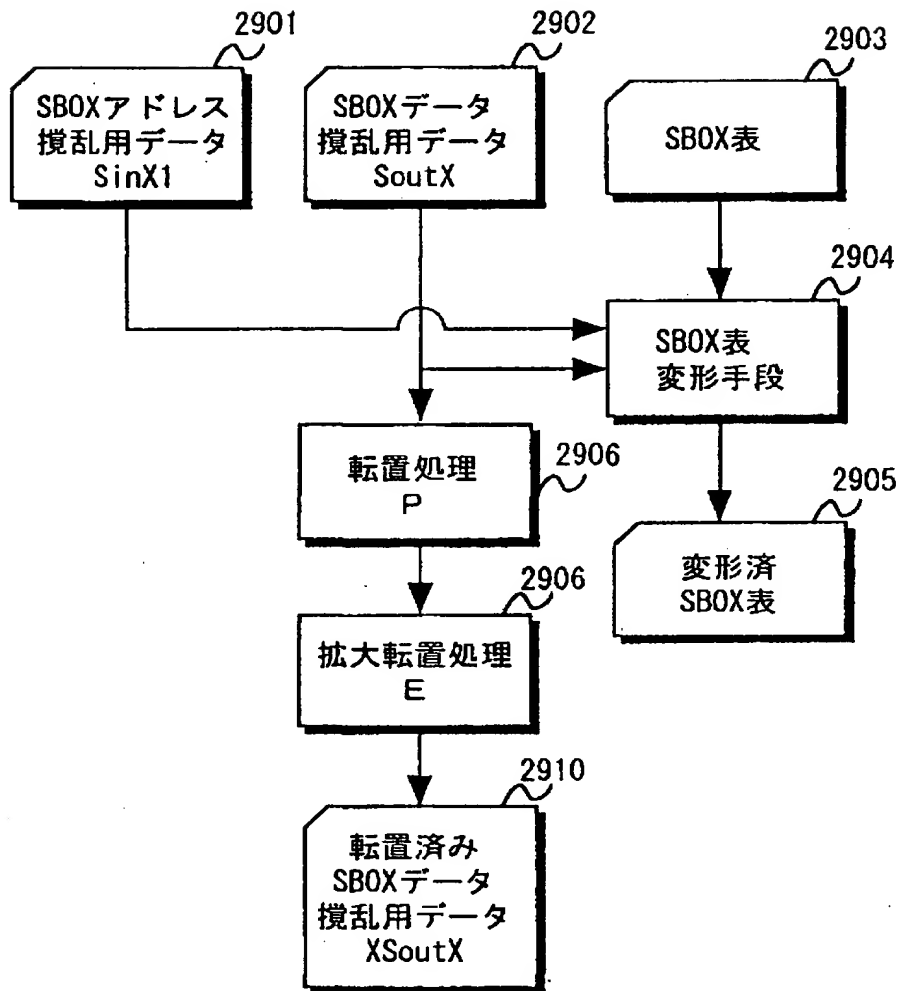
【図 28】

図 28



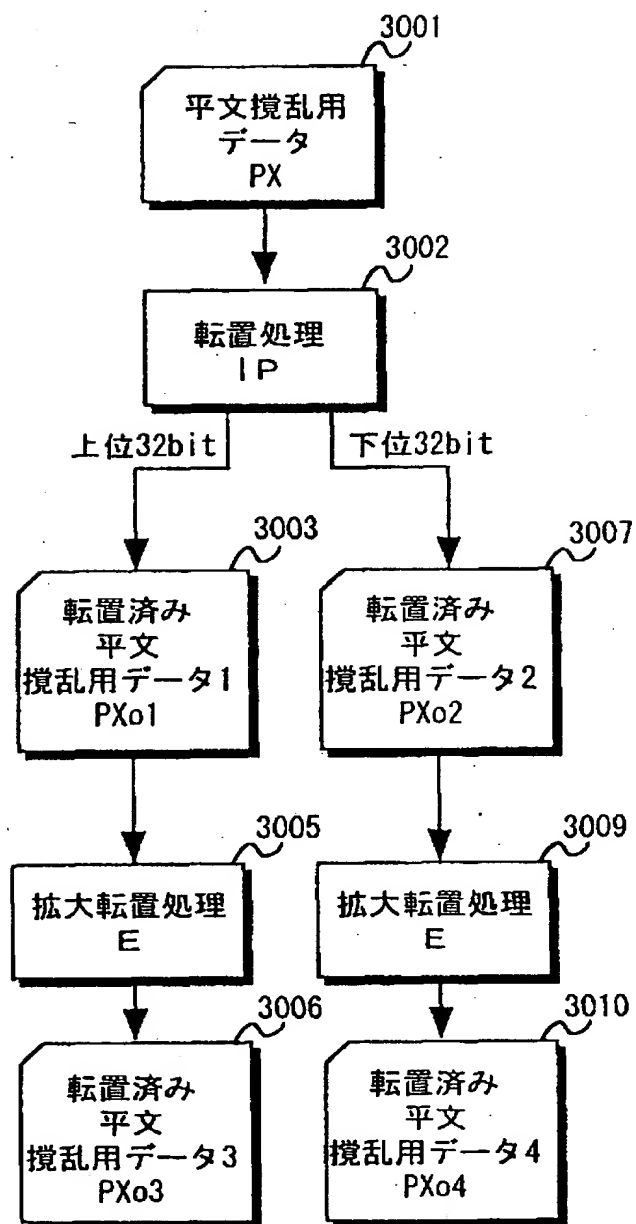
【図 2 9】

図 29



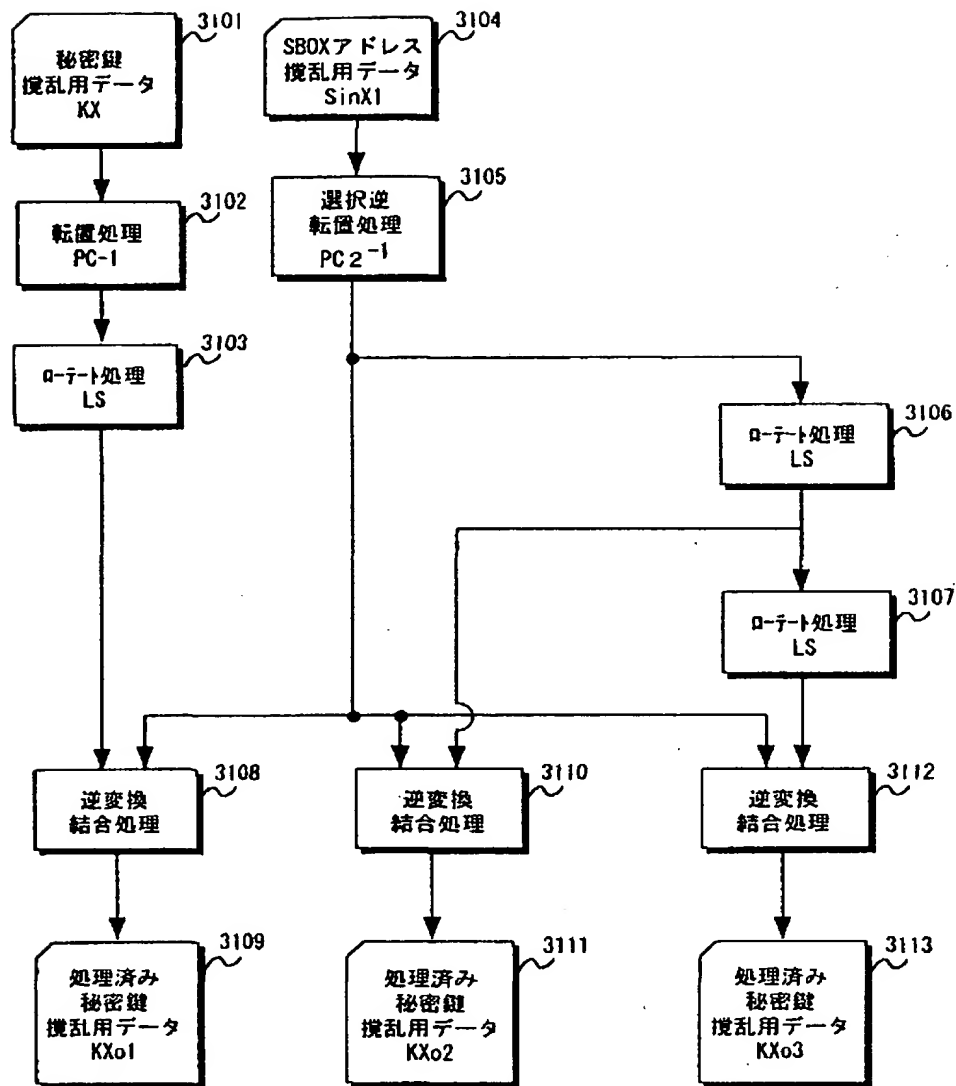
【図 30】

図 30



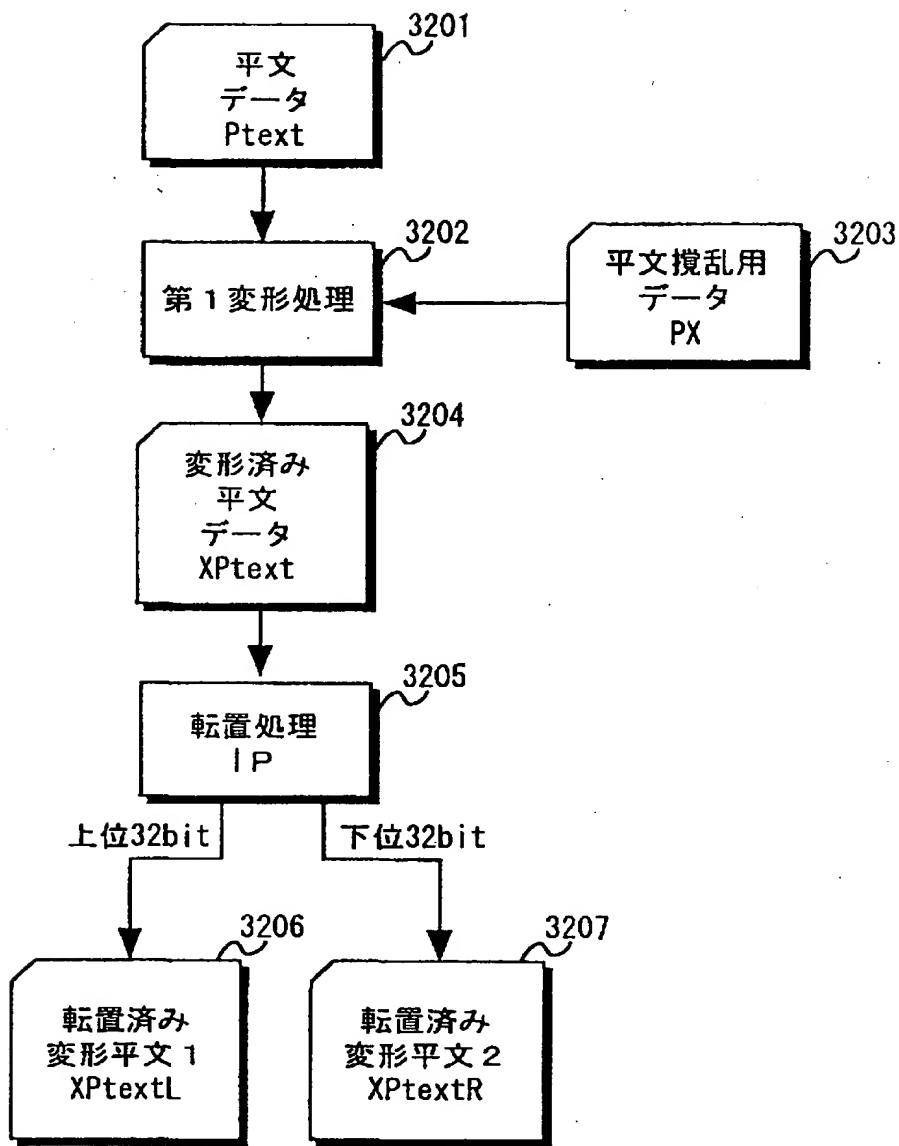
【図 31】

図 31



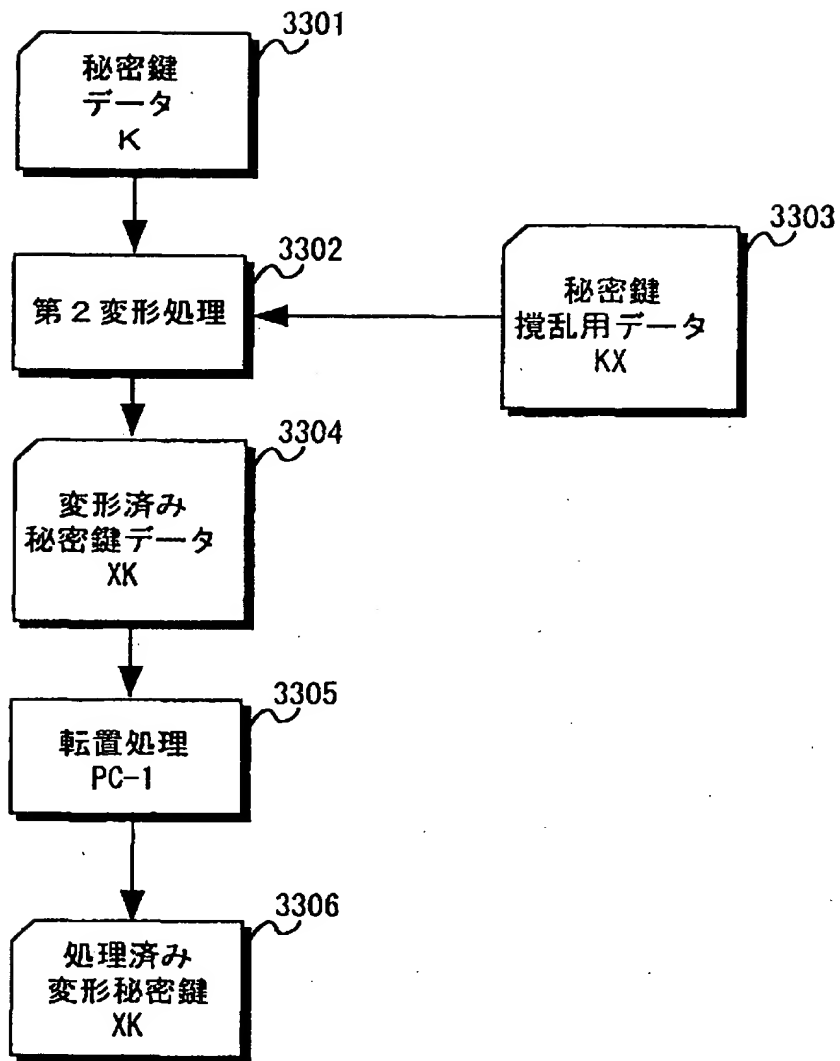
【図 3 2】

図 32



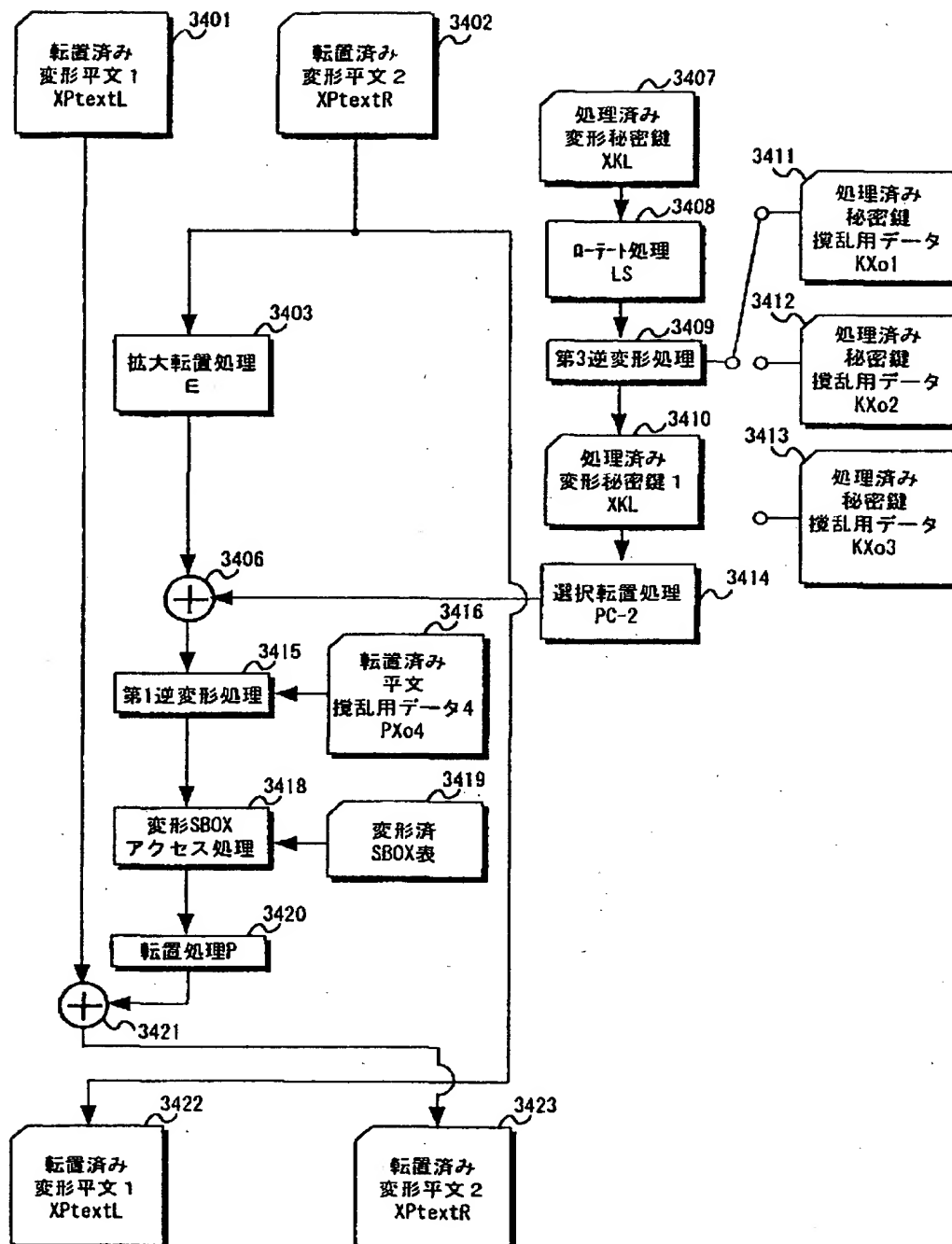
【図 3 3】

図 33



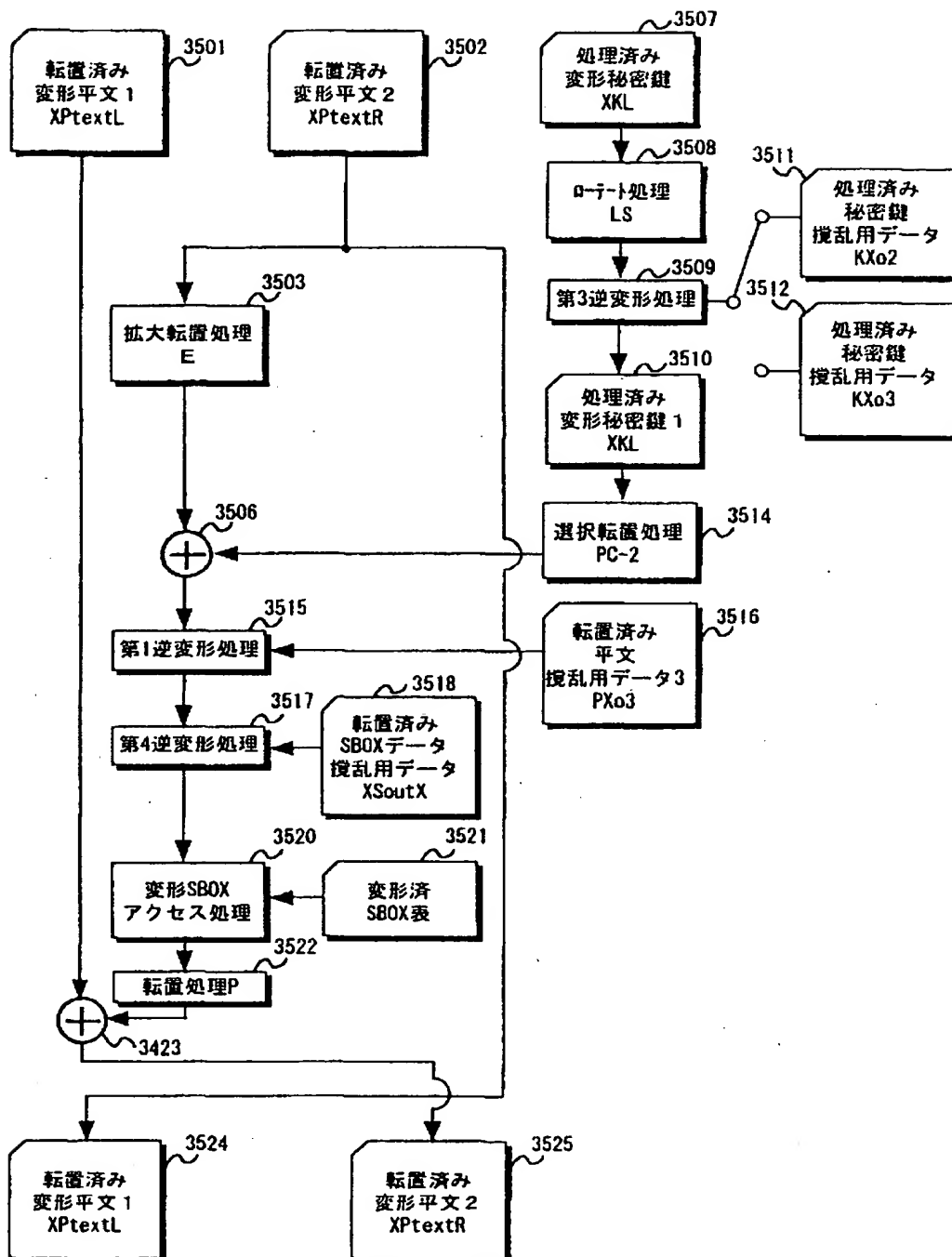
【図 34】

図 34



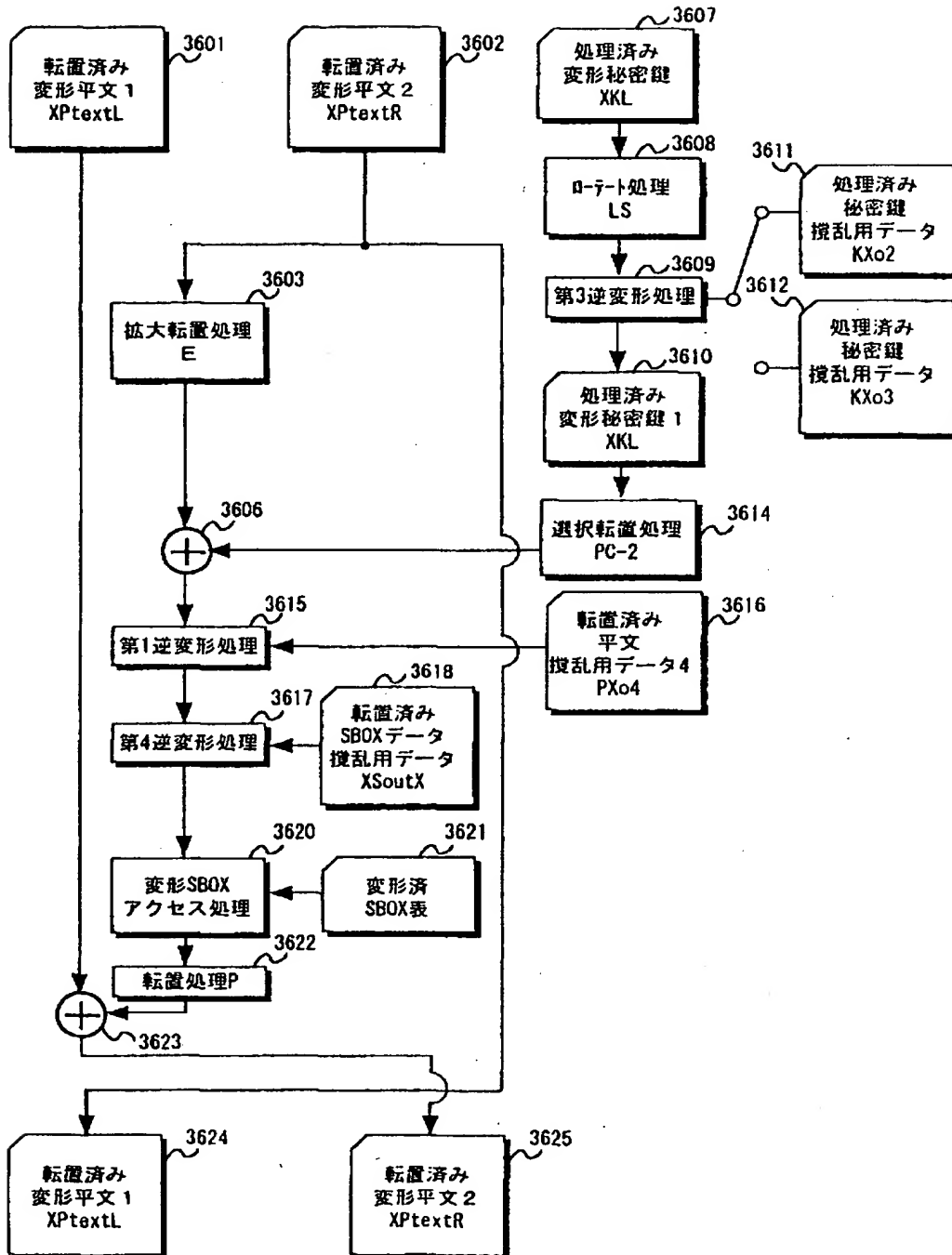
【図 35】

図 35



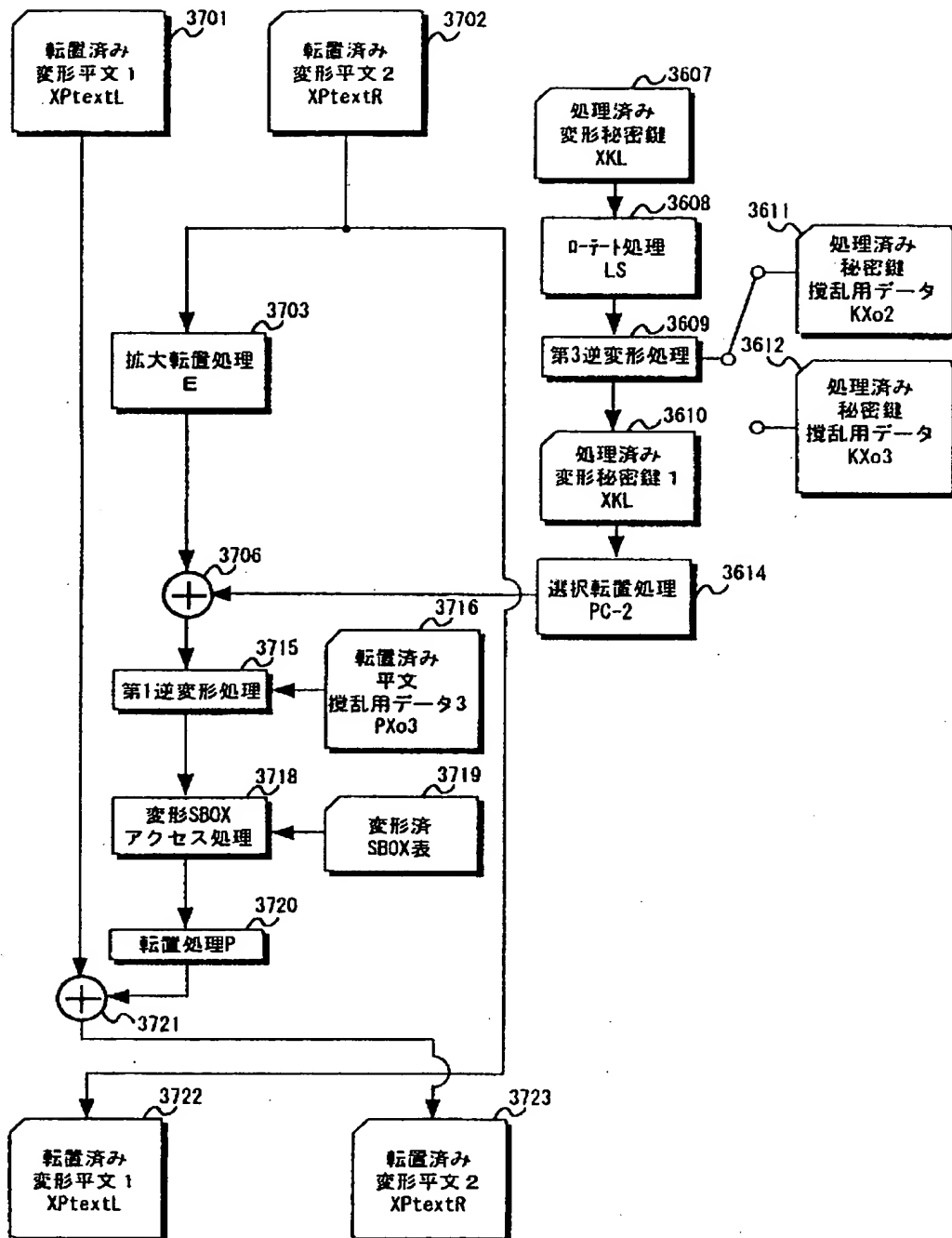
【図 36】

図 36



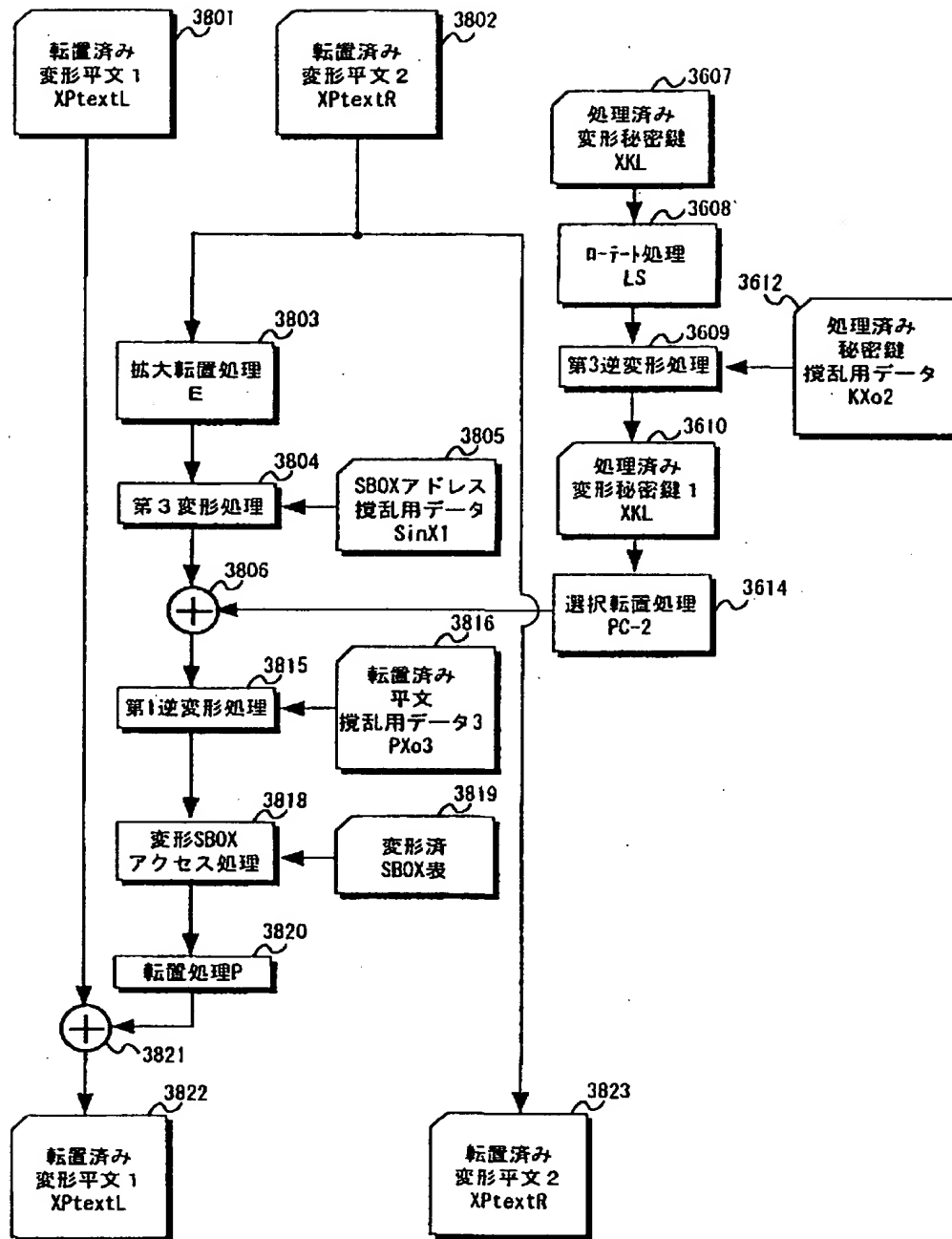
【図 37】

図 37



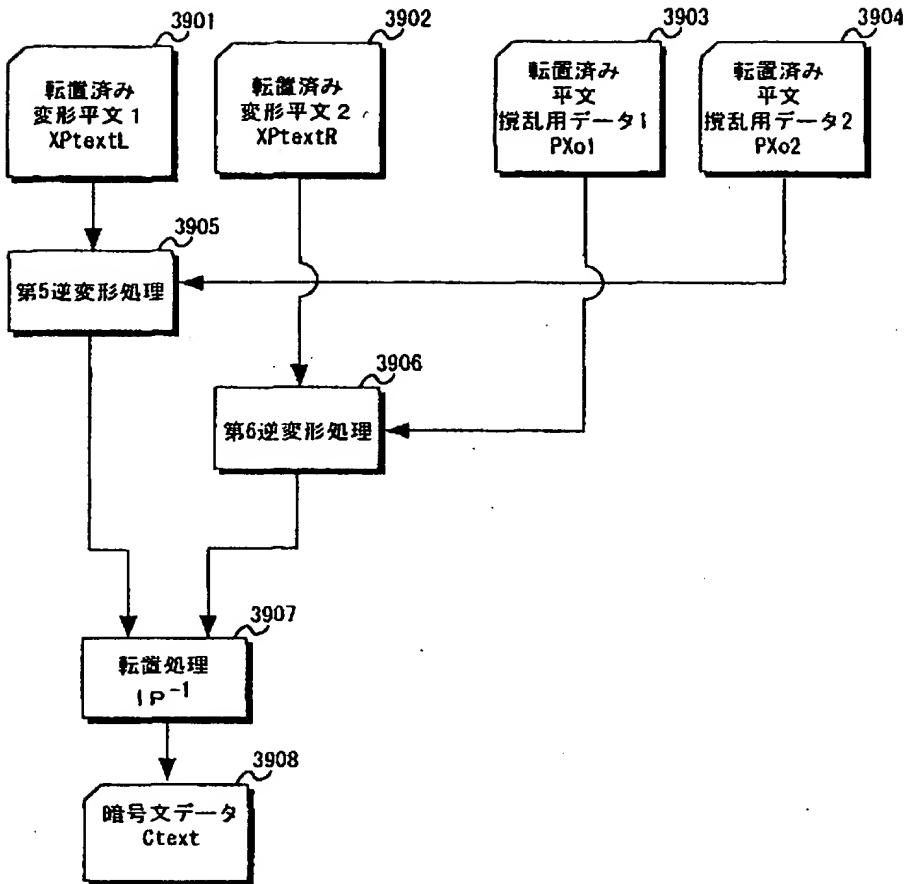
【図38】

図 38



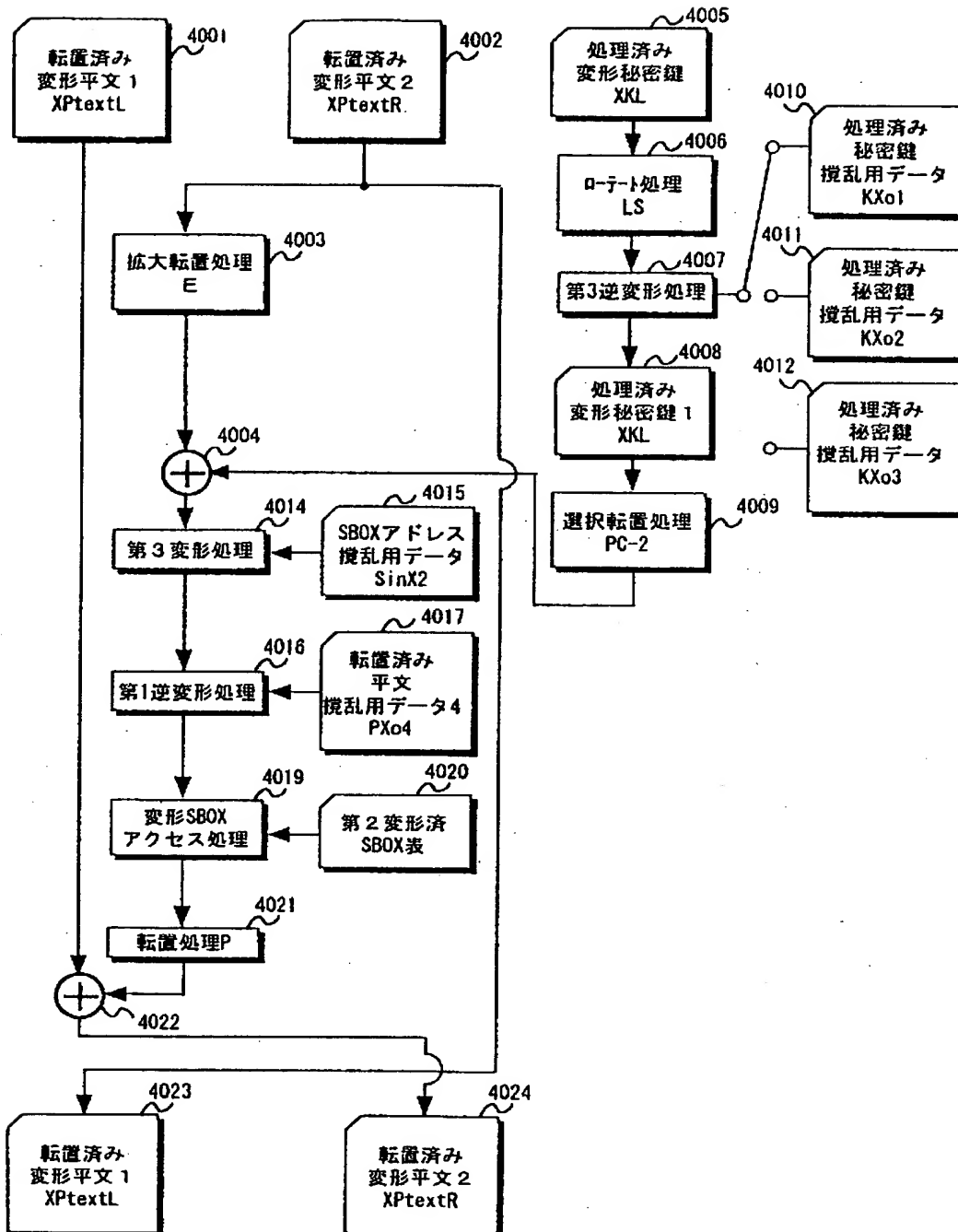
【図 39】

図 39



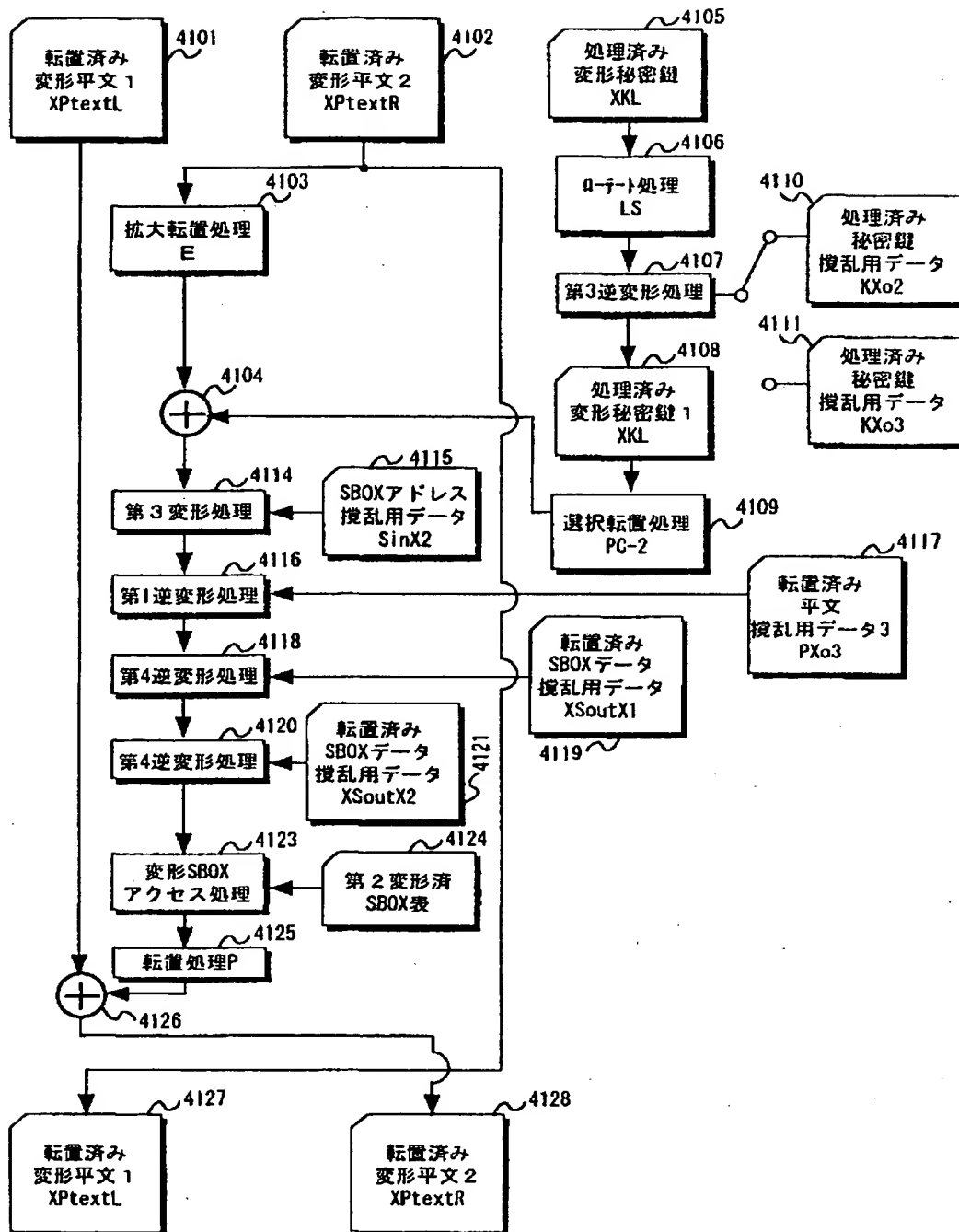
【図 40】

図 40



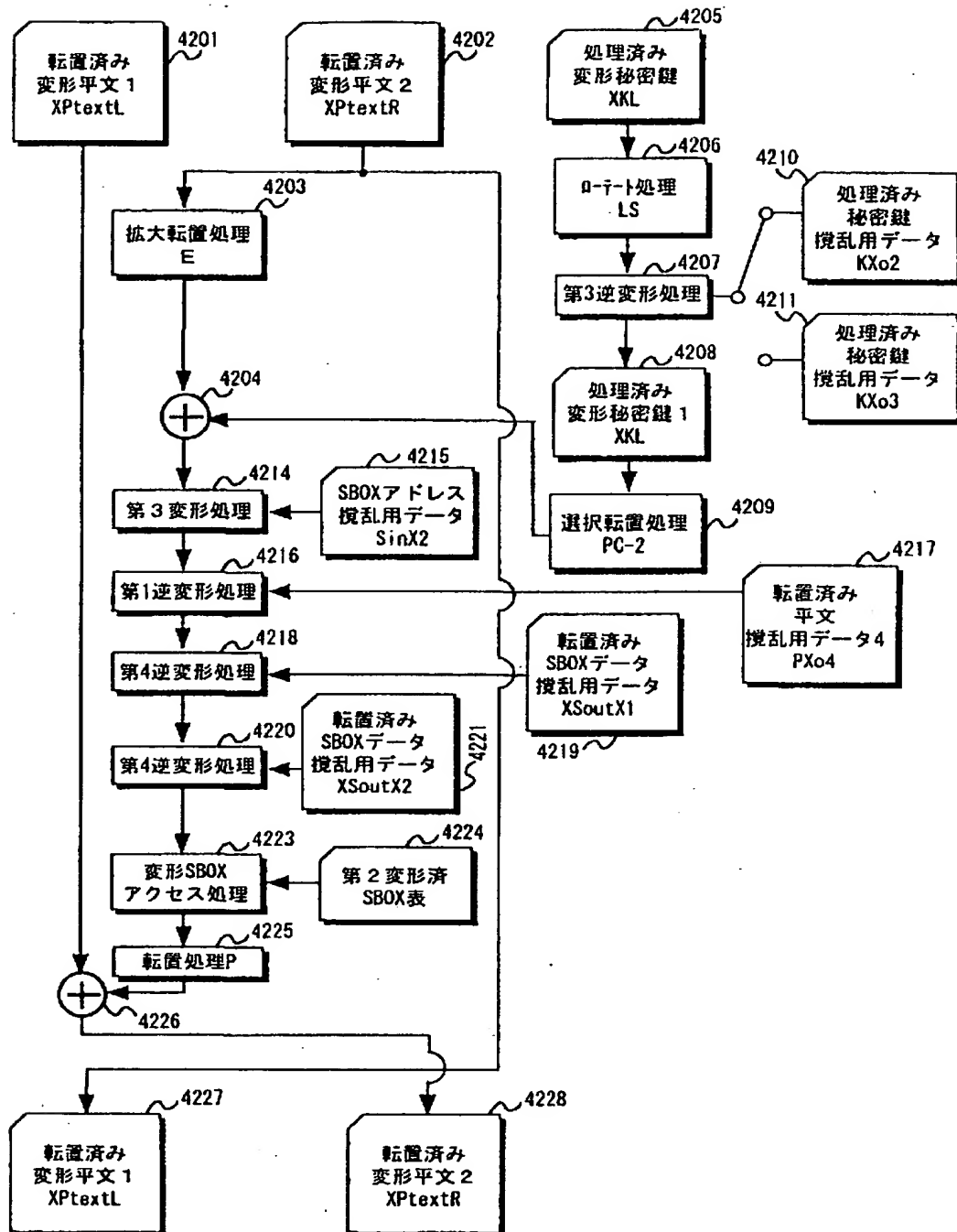
【図 41】

図 41



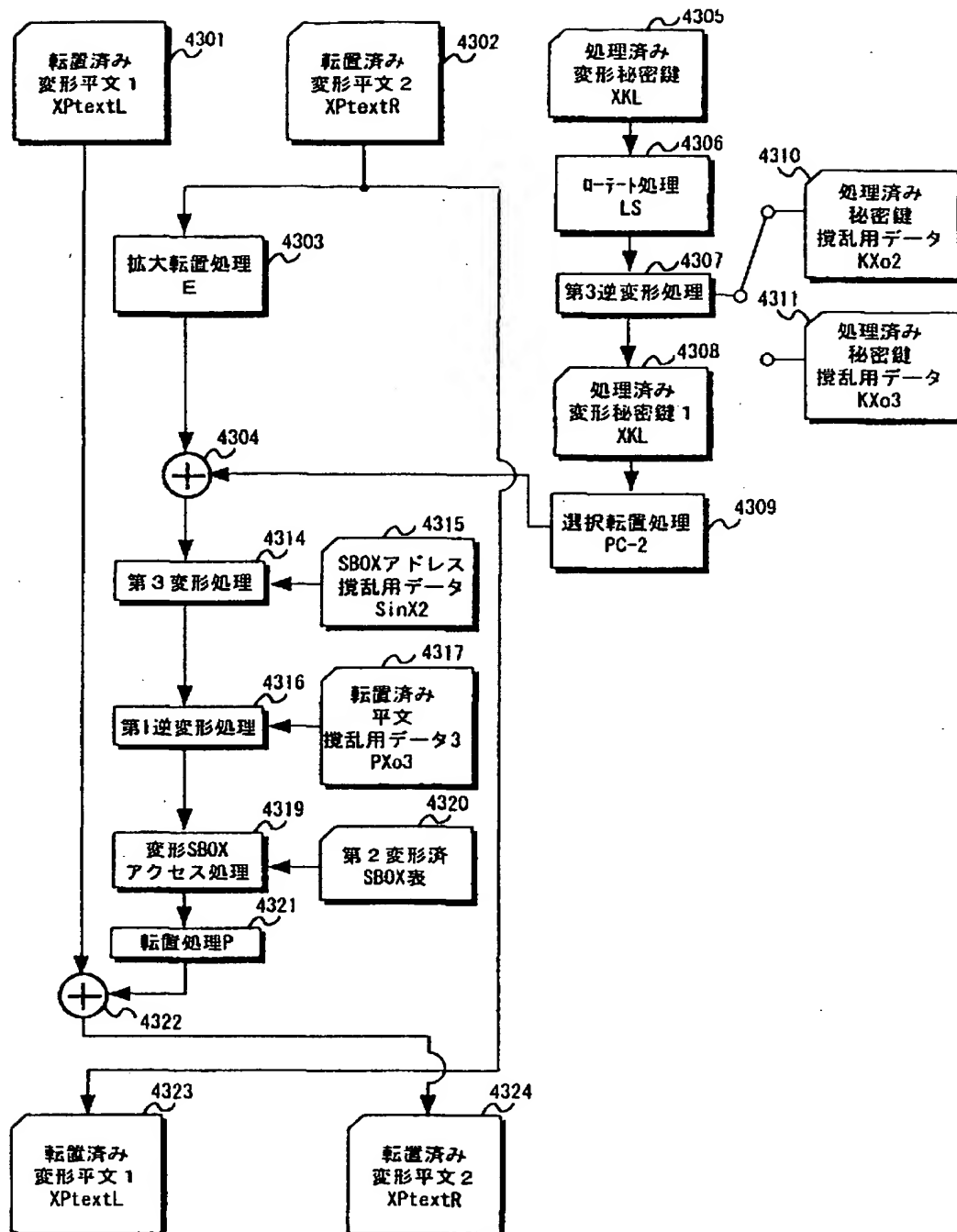
【図 4 2】

図 42



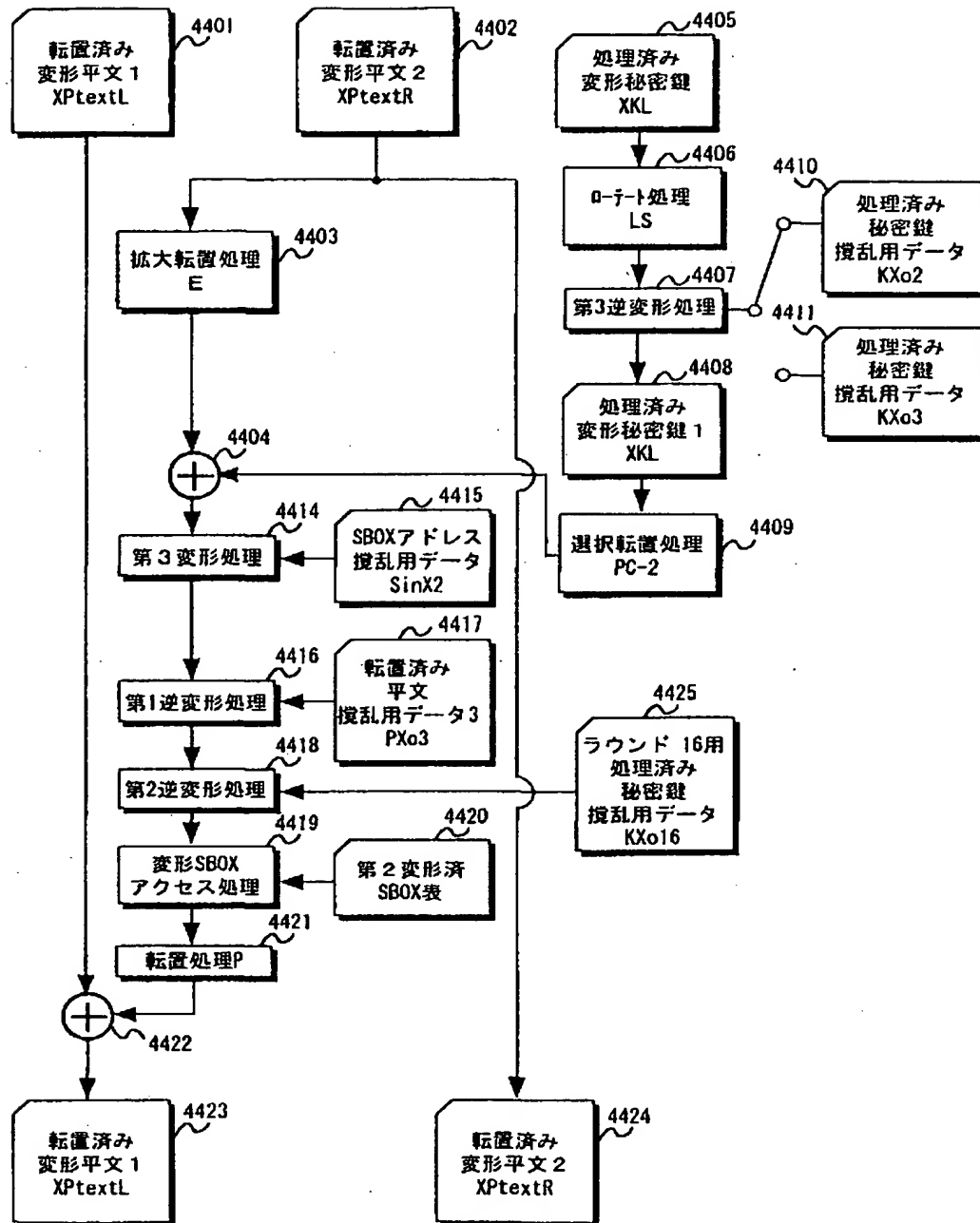
【図 43】

図 43



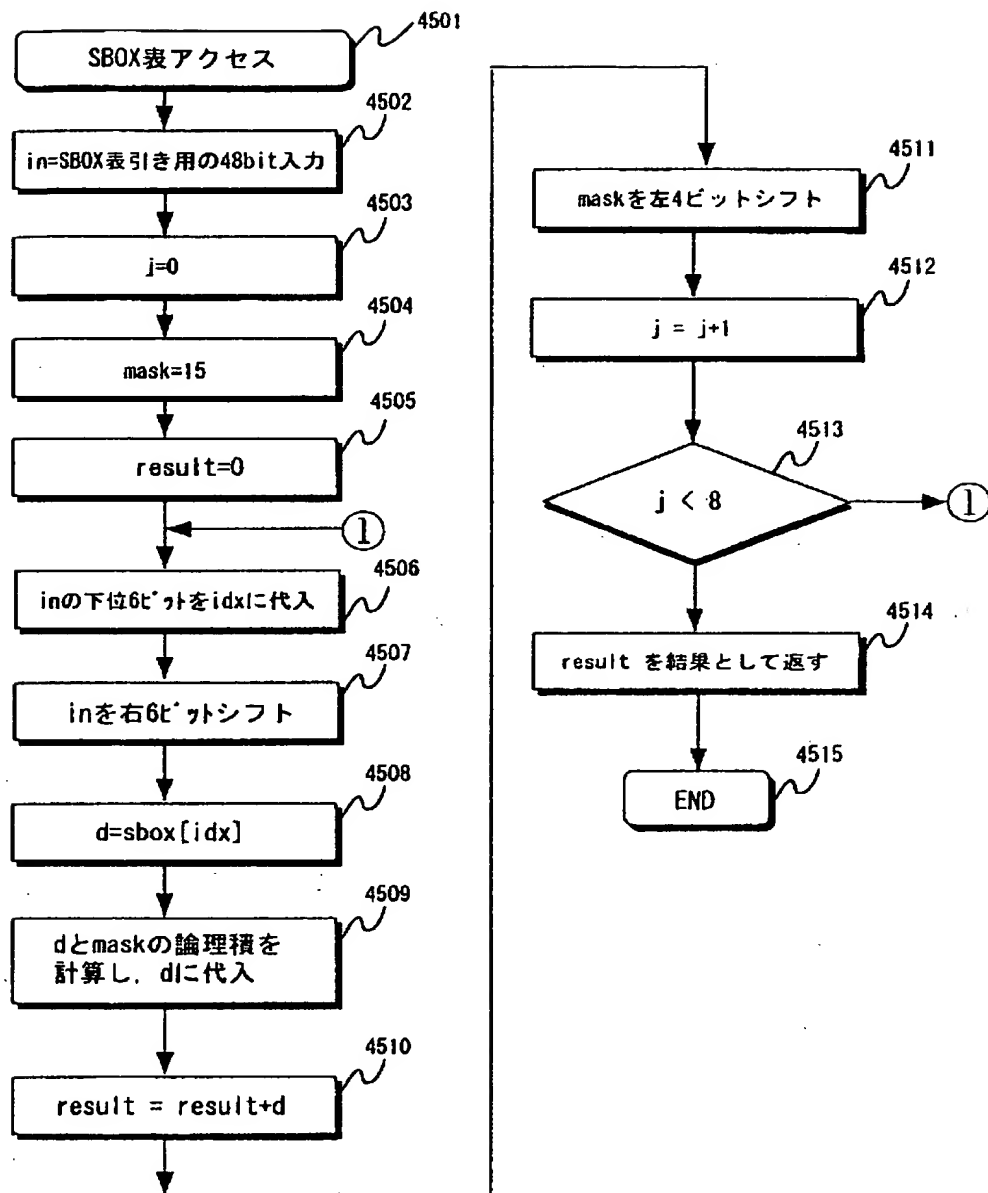
【図 44】

図 44



【図 45】

図 45



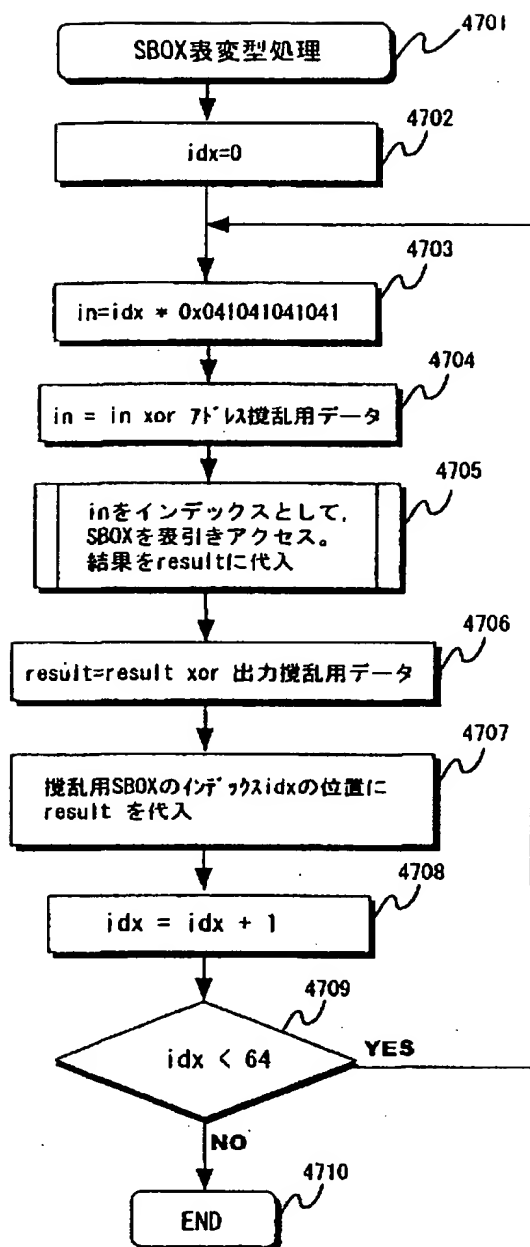
【図 4 6】

図 46

index	value	index	value	index	value	index	value
0	0xEFA72C4D	16	0x3911803A	32	0x40DA4917	48	0xF5BFF7A0
1	0x03DDEAD1	17	0xAC2456EC	33	0xFD13B462	49	0x5B49689F
2	0x410DC1B2	18	0xA7D25DC9	34	0x1E662E4B	50	0xC81190F6
3	0xFD78BF0F	19	0x60870135	35	0xC8AF83B1	51	0xB6F4FE5C
4	0xD89E4A28	20	0x62C83393	36	0xE7491FB4	52	0x9C23C46A
5	0x740B24BD	21	0xC152FD56	37	0x8AD0C2DE	53	0x37E50109
6	0x1EE31FE4	22	0xCD75F47E	38	0x8B90B5D1	54	0x76CE5A8D
7	0x4795C278	23	0xBAECAECB	39	0x21067C87	55	0xEC3B97F0
8	0x266079F6	24	0x5CBBDE55	40	0xDA8CA2C9	56	0x3955610F
9	0xEF36474A	25	0x69C13020	41	0x436A1914	57	0xA0BCA6E3
10	0xFB36A20F	26	0x904C07A0	42	0x64FBD83C	58	0xA3A23D53
11	0x224F7C93	27	0x59BA9BFE	43	0x9F91E54A	59	0x05574025
12	0xB3F9B68B	28	0x0524E56C	44	0x2D377C7E	60	0x52E80B95
13	0xD860D917	29	0x3BFE8389	45	0x148D2FA8	61	0x6E225836
14	0x845A68D1	30	0x7A8F9B17	46	0xB10D83E2	62	0x0F74E628
15	0x1EA315A4	31	0x85196862	47	0x7278DA7D	63	0xD9CE3DCB

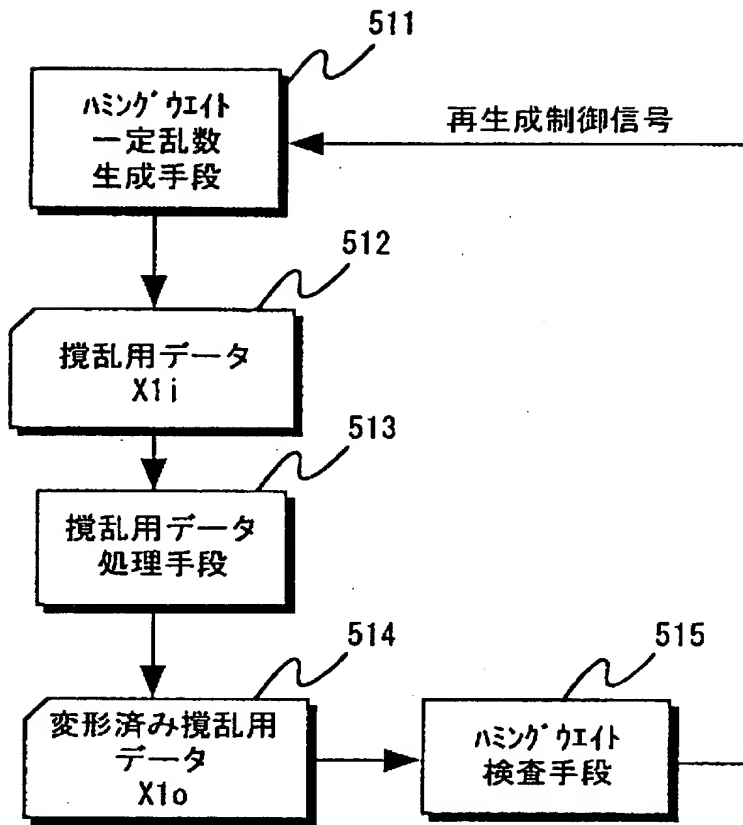
【図 47】

図 47



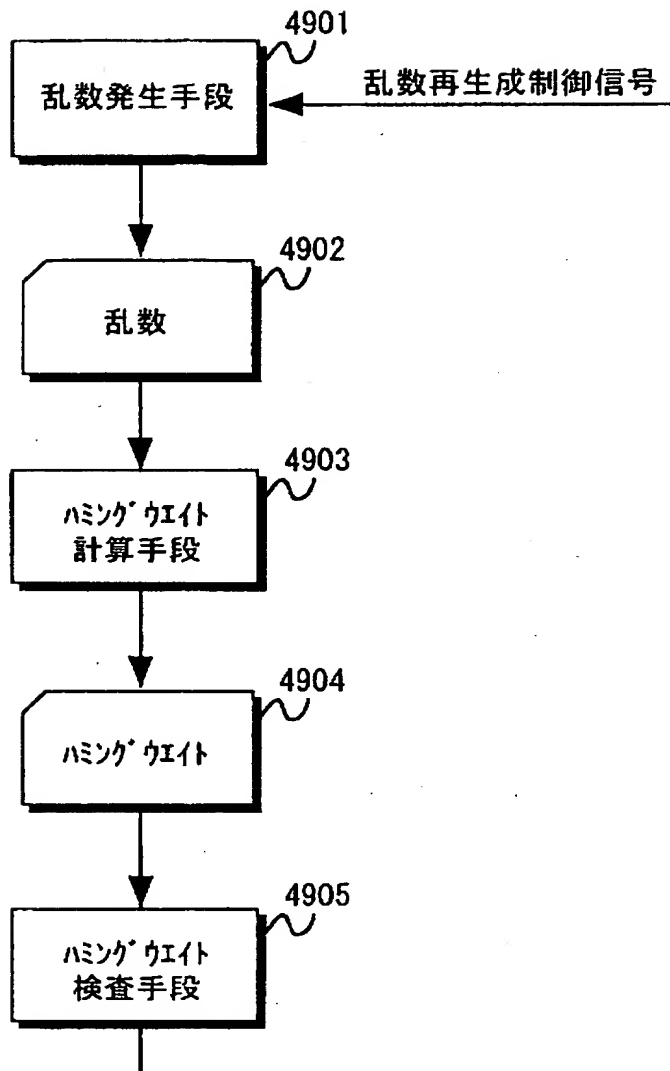
【図 4 8】

図 48



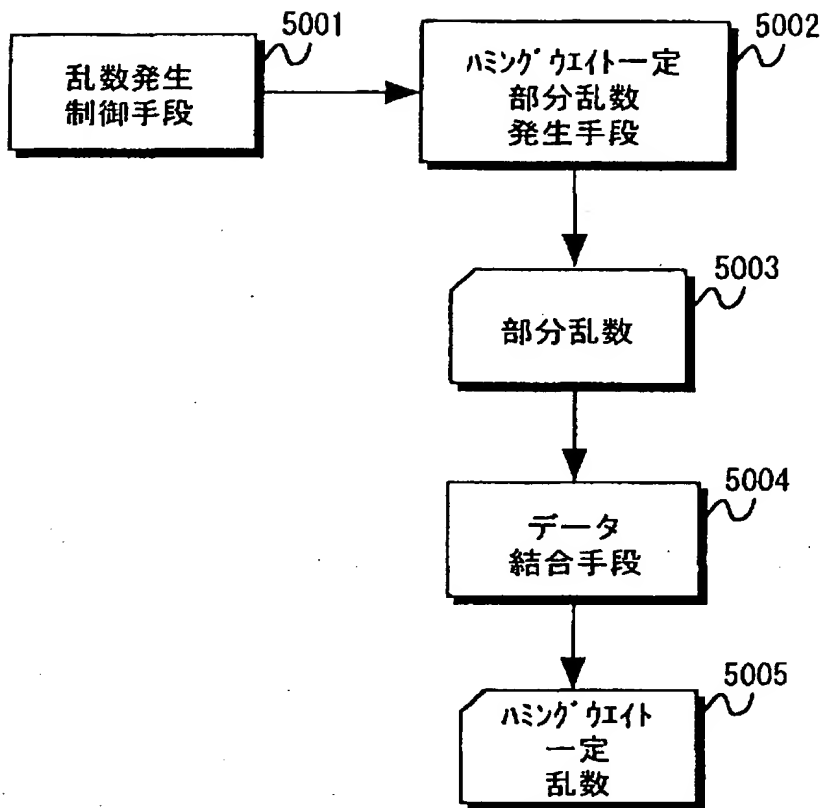
【図 4 9】

図 49



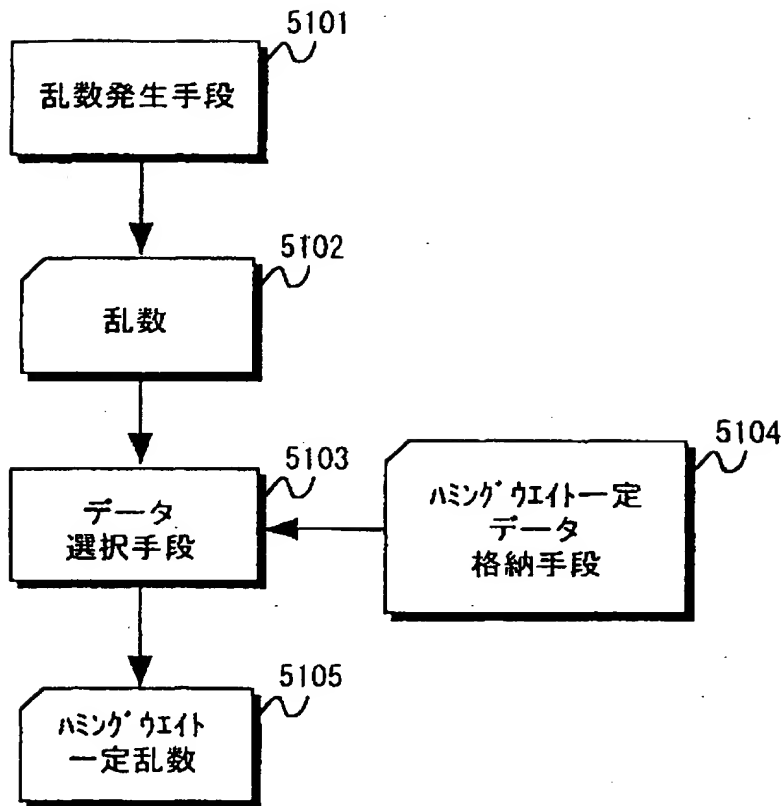
【図 5 0】

図 50



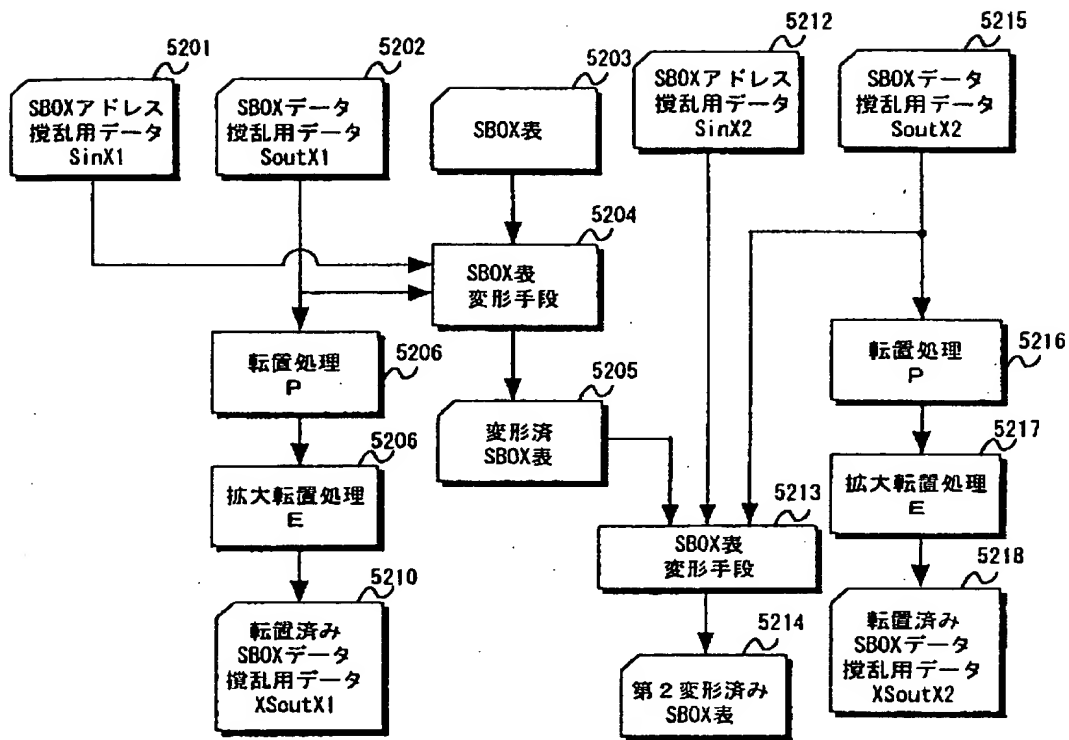
【図 51】

図 51



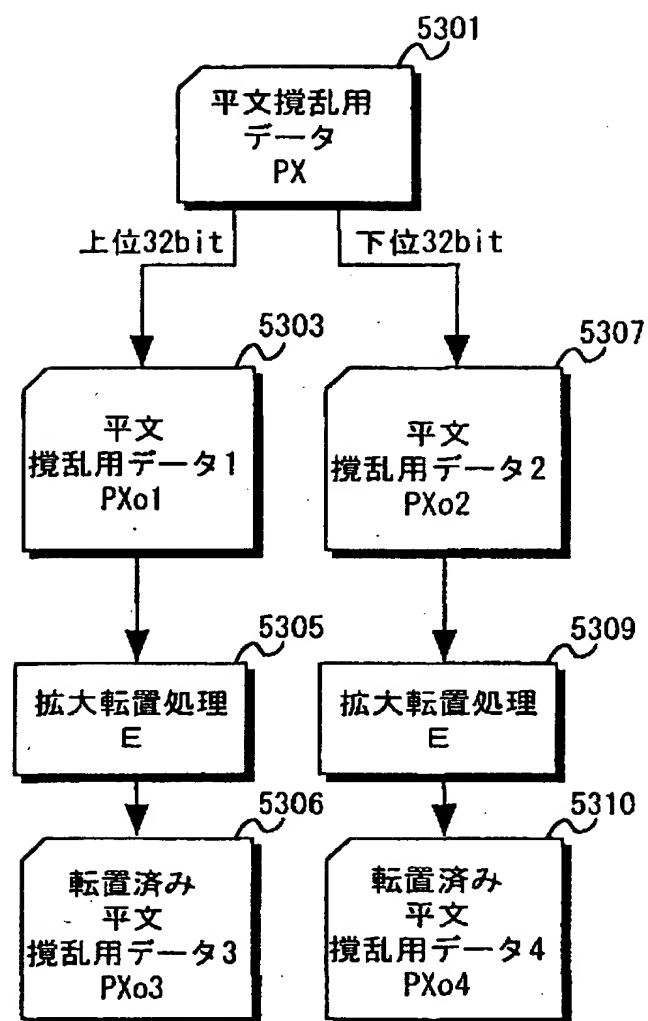
【図 52】

図 52



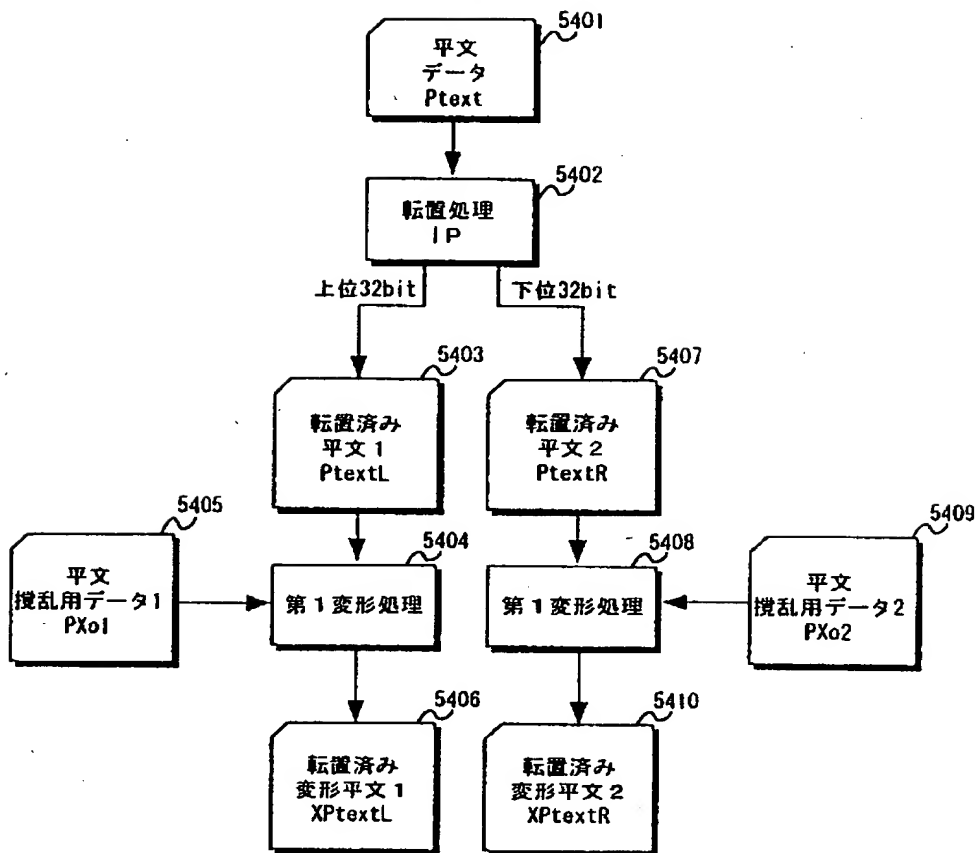
【図 5 3】

図 53



【図 54】

図 54



【図 55】

図 55

(データ処理が左ローテートの場合)

1201

X1i	X1o
0x55555555	0xaaaaaaaa
0x33333333	0x66666666
0x66666666	0xcccccccc

【図 5 6】

図 56

第1攪乱用	第2攪乱用
0x1c71c71c71c7	0x55555555
0x38e38e38e38e	0xAAAAAAAA
0x34d34d34d34d	0x0F0F0F0F
0x2cb2cb2cb2cb	0x71717171
0x38e38e38e38e	0x5a5a5a5a

【図 5 7】

図 57

第1攪乱用データ=0x1c71c71c71c7,
第2攪乱用データ=0x55555555 の場合

1507

Index	Value	Index	Value	Index	Value	Index	Value
0	0xEFA7204D	16	0x3911803A	32	0x40DA4917	48	0xF5BFF7A0
1	0x03DDEAD1	17	0xAC2456EC	33	0xFD13B462	49	0x5B496B9F
2	0x410DC1B2	18	0xA7D25DC9	34	0x1E662E4B	50	0xC81190F6
3	0xFD78BF0F	19	0x60870135	35	0x08AF83B1	51	0xB6F4FE5C
4	0xD89E4A28	20	0x62C83393	36	0xE7491FB4	52	0x9C23C46A
5	0x740B24BD	21	0xC152FD56	37	0x8AD0C2DE	53	0x37E50109
6	0x1EE31FE4	22	0xCD75F47E	38	0x8B90B5D1	54	0x76CE5A8D
7	0x4795C278	23	0xBAECAECB	39	0x21067C87	55	0xEC3B97F0
8	0x266079F6	24	0x5CBBDE55	40	0xDA8CA2C9	56	0x3955610F
9	0xEF36474A	25	0x96C13020	41	0x436A1914	57	0xA0BCA6E3
10	0xFB36A20F	26	0x904C07A0	42	0x64FBD83C	58	0xA3A23D53
11	0x224F7C93	27	0x59BA9BFE	43	0x9F91E54A	59	0x05574025
12	0xB3F9B68B	28	0x0524E56C	44	0x2D377C7E	60	0x52E80B95
13	0xD86D917	29	0x3BFE8389	45	0x148D2FA8	61	0x6E225836
14	0x845A68D1	30	0x7A8F9B17	46	0xB10D83E2	62	0x0F74E628
15	0x1EA315A4	31	0x85196862	47	0x7278DA7D	63	0xD9CE3DCB

1507

Index	Value	Index	Value	Index	Value	Index	Value
0	0x12C0972D	16	0xEF89FB9E	32	0x745329D2	48	0xB96EC2A5
1	0x4BB64AB1	17	0x9820A12B	33	0xDE05E084	49	0x239B0FD8
2	0x215E71E8	18	0x9407A803	34	0xDF85978B	50	0x62B0545C
3	0x8DCB1F7D	19	0x379D66C6	35	0xB21C4AE1	51	0xC976913F
4	0xA82DEA5A	20	0x35D25460	36	0x9DFAD6E4	52	0xE3A1AB09
5	0x145894E7	21	0xF287089C	37	0x4B337B1E	53	0x9D44C5A3
6	0x5688BF84	22	0xF97103B9	38	0xA846E137	54	0x0E1C3ECA
7	0xBAF27918	23	0x6C44D56F	39	0x158F1C42	55	0xA0EAA2F5
8	0x4BF640F1	24	0xD04C3D37	40	0x272D8F28	56	0x8C9B689E
9	0xD10F3D84	25	0x2FDACE42	41	0xE458D6B7	57	0x5A21B37D
10	0x8D358C42	26	0x6EABD6DC	42	0x41D87AFD	58	0x3B770D63
11	0xE6ACE3DE	27	0x5071B039	43	0x7862292B	59	0x07BD5EC0
12	0x771A29C6	28	0x0CEFCEAB	44	0xCAC4B01F	60	0x50021570
13	0xAE63F75A	29	0xC51952F5	45	0x31AE8D69	61	0xF6F76806
14	0xBA63121F	30	0xC3946575	46	0x163F4C41	62	0xF5E9F3B6
15	0x73352CA3	31	0x09EE8B00	47	0x8FD9F79C	63	0x6C00345A

【図 58】

図 58

1602

Index	Value	Index	Value	Index	Value	Index	Value	Index	Value
0	0x12C0972D	16	0xEF89FB9E	32	0x745329D2	48	0xB96EC02A5	第1攪乱用	0x1c71c71c71c7
1	0x48B64AB1	17	0x9820A12B	33	0xDE05E084	49	0x239B0FD8	第2攪乱用	0x55555555
2	0x215E71E8	18	0x9407A803	34	0xDF85978B	50	0x62B0545C		
3	0x8DCB1F7D	19	0x379D66C6	35	0xB21C4AE1	51	0xC976913F		
4	0xA82DEA5A	20	0x35D25460	36	0x9DFAD6E4	52	0xE3A1AB09		
5	0x145894E7	21	0xF287089C	37	0x4B337B1E	53	0x9D44C5A3		
6	0x5688BF84	22	0xF97103B9	38	0xA846E137	54	0x0E1C3ECA		
7	0xBAF27918	23	0x6C44D56F	39	0x158F1C42	55	0xA0EAA2F5		
8	0x4BF640F1	24	0xD04C3D37	40	0x272D8F28	56	0x8C9B689E		
9	0xD10F3084	25	0x2FDACE42	41	0xE458D6B7	57	0x5A21B37D		
10	0x8D358042	26	0x6EABD6DC	42	0x41D87AFD	58	0x3B770D63		
11	0xE6ACE3DE	27	0x5071B039	43	0x7862292B	59	0x078D5EC0		
12	0x771A29C6	28	0x0CEFC0EAB	44	0xCAC4B01F	60	0x50021570		
13	0xAE63F75A	29	0xC51952F5	45	0x31AE8D69	61	0xF6F76806		
14	0xBA63121F	30	0xC3946575	46	0x163F4C41	62	0xF5E9F3B6		
15	0x73352CA3	31	0x09EE8B00	47	0x8FD9F79C	63	0x6C00345A		

【書類名】 要約書

【要約】

【課題】 本願発明は、高いセキュリティを持つカード部材などの耐タンパー情報処理装置を提供するものである。技術的な課題は、カード部材、例えばＩＣカード用チップでのデータ処理と消費電流との関連性を減らすことである。

【解決手段】 本願発明の着眼点は、ＩＣカード用チップで消費される電流値と、処理されているデータの関連性を減らすための方法として、処理するデータを攪乱用データで変形し、データの処理を変形したデータで処理し、処理後に攪乱用データを用いて逆変換し、正しい処理結果を求めるものである。

【選択図】 図 4

特2001-046250

出 願 人 履 歴 情 報

識別番号 [000005108]

1. 変更年月日	1990年 8月31日
[変更理由]	新規登録
住 所	東京都千代田区神田駿河台4丁目6番地
氏 名	株式会社日立製作所